NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

A DATABASE APPLICATION FOR THE NAVAL COMMAND PHYSICAL READINESS TESTING PROGRAM

by

Frances M. Quinones

December 1998

Thesis Advisor: C. Thomas Wu

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 1998	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE: A DATABASE APPLICATION FOR THE NAVAL COMMAND PHYSICAL READINESS TESTING PROGRAM	5. FUNDING NUMBERS

6. AUTHOR(S) Quinones, Frances M.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT: Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE:

13. ABSTRACT *(maximum 200 words)*

IT21 envisions a Navy with standardized, state-of-art computer systems. Based on this vision, Naval database management systems will also need to become standardized among Naval commands. Today most commercial off the shelf (COTS) database management systems provide a graphical user interface. Among the many Naval database systems currently in use, the Navy's Physical Readiness Program database has continued to exist at the command level in a DOS version. This version is outdated since it does not reflect current regulations and lacks many capabilities desired by training coordinators.

A central question in the design of a graphical user interface is the need to create an application program which will allow Physical Readiness Coordinators at Naval commands to access the Physical Readiness Training database for local personnel in a windows-based environment without the need of a Database Management System environment (such as Microsoft Access).

The design recommendations and implementation of a prototype are the primary research areas for this thesis.

14. SUBJECT TERMS Database			15. NUMBER OF PAGES 120
			16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

A DATABASE APPLICATION FOR THE NAVAL COMMAND PHYSICAL READINESS TESTING PROGRAM

Frances M. Quinones

Lieutenant, United States Navy

B.A., San Diego State University, 1989

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL

December 1998

Author: _____

Frances M. Quinones

Approved by: _____

C. Thomas Wu, Thesis Advisor

_____

Chris Eagle, Second Reader

_____

Dan Boger, Chairman, Department of Computer Science

iv

ABSTRACT

IT21 envisions a Navy with standardized, state-of-art computer systems. Based on this vision, Naval database management systems will also need to become standardized among Naval commands. Today most commercial off the shelf (COTS) database management systems provide a graphical user interface. Among the many Naval database systems currently in use, the Navy's Physical Readiness Program database has continued to exist at the command level in a DOS version. This version is outdated since it does not reflect current regulations and lacks many capabilities desired by training coordinators.

A central question in the design of a graphical user interface is the need to create an application program which will allow Physical Readiness Coordinators at Naval commands to access the Physical Readiness Training database for local personnel in a windows-based environment without the need of a Database Management System environment (such as Microsoft Access).

The design recommendations and implementation of a prototype are the primary research areas for this thesis.

v

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

I. INTRODUCTION

A. BACKGROUND

In March 1997, Admiral Archie Clemins, CINCPACFLT, announced the IT-21 program, promulgating hardware and software implementation standards for the 21st century. It dictated the use of relational databases that employ web technology in accordance with a common operating environment (Oracle, Sybase, SQL server, Access, etc.) to support data requirements and application development. Furthermore, it emphasized that "all process engineering initiatives that result in design/redesign of a data collection/capture system must use common operating environment compliant relational database management systems (RDBMS) software [Ref. 1]."

Admiral Clemins further characterized IT-21 as a "paradigm shift in how we create, manage, and retrieve information, and it allows us to reengineer our processes, which is critical in this era of reduced budgets and geopolitical uncertainty. [Ref. 2]." These processes include the way in which the Navy maintains logistics, personnel, training, medical and supply information.

The Navy's Physical Readiness Program is one such process involving both personnel and training. Even though today most commercial off the shelf (COTS) database management systems utilize a graphical user interface, the Navy's Physical Readiness Program database has continued to exist at the command level as a DOS based application. This outdated system is not providing the functionality necessary to meet the administrative demands placed on the command and more specifically on the Command Fitness Coordinator (CFC).

1

There exists a need to develop a better process which will not only meet IT21 requirements and comply with the recently published Physical Readiness Program instruction (OPNAVINST 6110.1E), but will also provide a standard in the fleet and serve as the first step towards Navy-wide centralized network accessible Physical Readiness Program database.

B. DESCRIPTION OF THE NAVY'S PHYSICAL READINESS PROGRAM

In March 1998, the Chief of Naval Operations promulgated OPNAVINST 6110.1E, providing revised policy and guidance for the implementation of the Navy's Physical Readiness Program. [Ref. 3] In order to design the database application program, an understanding of the revised policy is paramount.

The instruction requires all Naval commands to conduct the Physical Readiness Test (PRT) twice a year. All Naval personnel must participate unless granted a waiver. In a change to the previous policy, special PRTs are allowed to be conducted on a case by case basis for personnel whose failure of the regular PRT could cost them advancement or other opportunities. Although the Commanding Officer is ultimately responsible for the program, the Command Fitness Coordinator (CFC) is responsible for administering the test and maintaining a record of each member's results.

The PRT is an evolution that starts with its announcement and culminates with the event itself. The PRT must be announced 12 weeks prior to its scheduled date. During this time and prior to the PRT, the CFC will ensure that personnel are physically ready to participate in the PRT.

1. Medical Requirements

Prior to taking the PRT, personnel must meet certain medical standards. The CFC maintains a Risk Factor Screening/Physical Readiness Test Result form, OPNAVINST 6110/2, better known as the "pink folder" for every member at his command. A history of prior PRTs is recorded in the pink folder and must be maintained for four years.

The CFC must verify that all members have a current 5-year physical. If the 5-year physical is not current, the member will not be allowed to participate in the PRT. In addition, personnel must complete a Risk Factor Questionnaire, regarding their present physical condition. Answering "yes" to any of eight questions will result in referral to medical and denial of participation in the PRT unless medically cleared by a physician. Medical waivers for a portion of the PRT or the entire PRT are permissible and the PRT is not considered a failure. However, three consecutive medical waivers require medical administrative action.

Following initial screening and ten days prior to the PRT, the CFC must weigh all members. A member must meet the height and weight standards outlined in OPNAVINST 6110.1E. If the member fails to meet the height and weight standards, he will be measured to determine his body fat. Neck, chest and waist measurements will be taken in males, and neck, waist and hips will be measured in females. Measurements will be logged in the pink folder. Based on these measurements, a computation will be performed and its result checked against a table in OPNAVINST 6110.1E to determine if the member is not within body fat standards. A marking of WS (within standards) or NS (not within standards) will be recorded. Pregnant members will be waived from measurements.

Members who are not within standards will be referred to medical, and if cleared by medical will be placed on a physical conditioning program. Members who fail the PRT must also participate in the physical conditioning program. If after 6 months participation in the conditioning program, the member has failed to progress, he will be referred to medical for participation in the BUMED-approved Weight Management Program. To participate in the program the member must meet the criteria set in Enclosure (5) of OPNAVINST 6110.1E. Graduates of this program must meet the body composition standards at the completion of the 1-year follow-up or they will be processed for administrative separation from the Navy.

2. Physical Readiness Test

The PRT consists of four events: sit-and-reach (Pass/Not Pass), run or swim (points per finish time), sit-ups (points per number completed), and push-ups (points per number completed). The points awarded are listed in a table in OPNAVINST 6110.1E, which takes into consideration age and sex.

There are four levels that can be achieved in all of the point-based portions of the PRT: Outstanding, Excellent, Good and Satisfactory. In addition, an overall score will be computed by adding the individual event scores. A failure in any single event with the exception of the sit-and-reach constitutes a failure of the PRT. Furthermore, completing the minimum requirements to pass each event does not guarantee passing the PRT. A minimum point total must be achieved to pass the PRT. If three PRT failures occur in a 4-year period, the member must be processed for administrative separation or denial of reenlistment.

3. Documentation Requirements

The CFC has many documentation requirements to properly administer the program. Enclosure (7) of OPNAVINST 6110.1E provides samples for several of the required forms. A Page 13 entry in enlisted member service records must be completed for PRT failures. The instruction provides a sample Page 13 for first and second time failures and another for third time failures. Also included in the instruction are samples for a Letter of Notification for officers who fail the PRT and a Command Physical Readiness Test Summary to be submitted annually to the Bureau of Naval Personnel.

Not included in the instruction but required are letters to the Bureau of Naval Personnel regarding PRT failures for officers and enlisted, and a referral to a medical officer for screening as possible candidates for a Bureau of Medicine approved weight management program.

In addition to forms included in the instruction, Naval commands develop their own forms to perform the following functions:

1. PRT timetable and checklist.

2. Measurements form.

3. PRT scoring sheet.

4. Assignment of members to the physical conditioning program upon failure.

5. Awards Certificates.

6. PRT summary sheet listing body fat standards and pass or fail for each member to be included in Evaluations and Fitness Reports.

7. PRT summary sheet listing total points and categories achieved for all command members.

The non-required forms vary from command to command. It is possible that some commands will have other forms tailored to their specific situations.

C. RESEARCH GOAL

The goal of this thesis is to develop a better method for CFCs to maintain the PRT information and reduce administrative demands while meeting IT-21 and recently published OPNAVINST 6110.1E requirements. This thesis envisions a solution with a Windows graphical user interface.

With this goal in mind, the desired results were many. The program had to provide an easy to use interface to an underlying database, prevent the need to learn a Database Management System (DBMS) application, distribute and install at low cost, provide a wide range of tailored statistical data and reports, reduce the administrative burden of the CFC, provide rapid individual or command status, and import existing PRT data to encourage adoption of the new program.

II. REVIEW OF EXISTING PRT DATABASE SYSTEMS

A. PRT APPLICATION - 1988 DOS VERSION

In May 1988, a PRT Software Package became available to the Fleet. Designed by ET1 William L. Brown, while on active duty in the U.S. Navy at FACSFAC San Diego, California, the program became the standard tool for managing PRT data for Naval commands and is still in use today.

1. Hardware and Software Requirements

The program was designed using TURBO PASCAL 4.0 for an 8088 or 80286 central processing unit with MS-DOS 3.10. Attempting to maximize 1988 technology, PRT cycles are saved as binary database files. Each personnel record requires 181 bytes of random access memory (RAM) with a maximum of 2,800 records for 640K of RAM. While records are being added, the remaining RAM is displayed and an error message will be generated if the user attempts to save more records than RAM available. Additionally, the program was designed to print forms using an IBM compatible, dot matrix printer and provided for saving databases to a 5.25-inch floppy disk. Although the package provided an installation file for ease of program installation, the printer must be configured through a series of DOS commands and utilities. [Ref. 4]

2. Program Design

The Software Package consists of five executable files: PRT.exe, Main.exe, Summary.exe, Forms.exe, and Merge.exe. PRT.exe is used to control all other software programs contained in the package. The user enters "PRT" at the DOS prompt to execute

7

the program. Main.exe is the PRT database program. It constructs all PRT databases and allows printouts of PRT results by various categories. Summary.exe generates, prints and saves to disk the Command PRT Summary Report. Forms.exe provides generic examples of Page 13 forms, Letters of Accomplishment, and others. Finally, Merge.exe merges databases within the same exam cycle into a single command database. This feature is provided for large commands where each department may have its own CFC, each having a departmental PRT database.

Upon execution of the program, the first menu, depicted in Figure 1, lists six selections available with function keys. The user may elect to start the database program, access the user's manual, command summary, and forms or merge several databases. The "Quit" option returns the user to the DOS prompt, exiting the program.
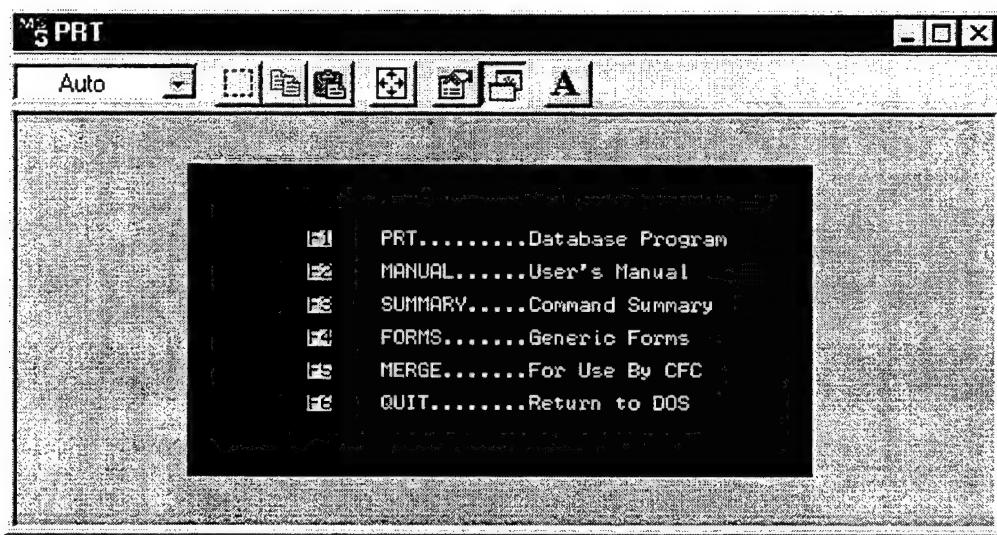
Figure 1. Initial Menu

If the user selects F1 in the initial menu, the menu, depicted in Figure 2, allows him to select either disk drive A or C and whether to create a new database, access an existing database, template a database, or merge a database. If EXISTING/MERGE DATABASE is selected, a directory listing of all existing PRT databases on the selected

8

drive is displayed. If either CREATE A DATABASE or TEMPLATE DATABASE is selected, the user is required to enter a name for the new PRT database.
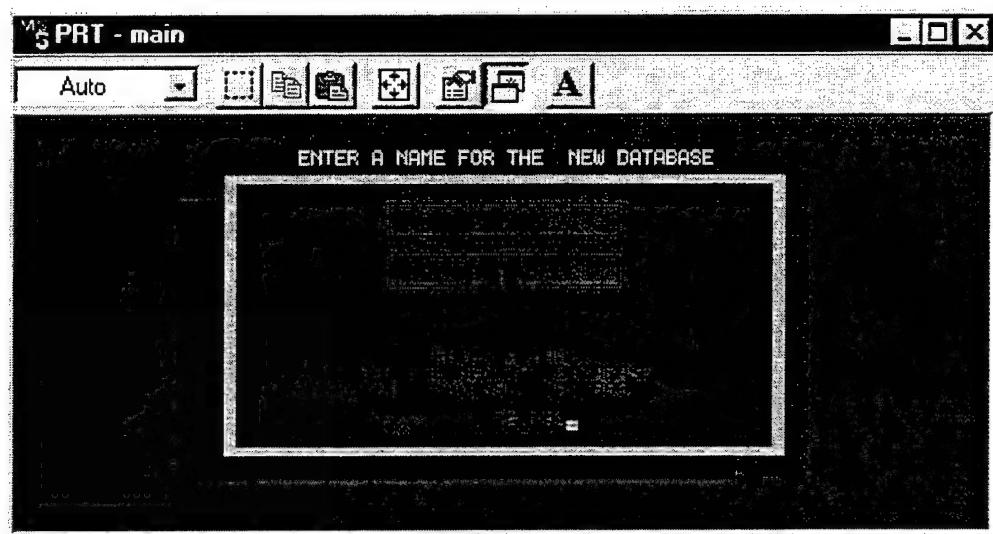
Figure 2. Database Selection Screen

After creation of the database name from the Database Selection Screen, the user is presented a new menu, Figure 3, to perform several actions on the selected database. The user can add a record to the current PRT database, change data within a record, delete a record, find a record by last name, list records by several categories, save all current changes to the database, input another PRT database and quit the program while saving the current changes.
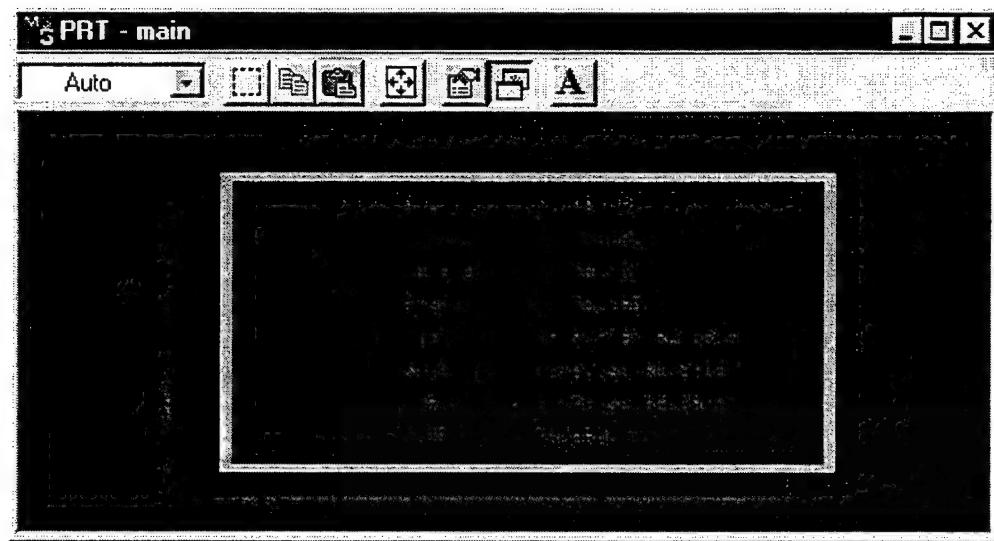
Figure 3. Record Action Selection Screen

3. Functionality

Attempting to alleviate the "unnecessary man-hours involved with recording PRT data and with deriving pertinent information from the collected data [Ref. 4]," the program provides the minimum functionality necessary to support the 1988 PRT program while easing the administrative demands placed on the CFC.

The creation of a database assists the CFC by having information available at his fingertips. This database consists of records, such as the one depicted in Figure 4, listing the member's personal information, test date and score. To create a record, the user inputs the last name, first name, rate, social security number (SSN), department, division, test date, birth date, sex, measurements, and test scores. The program automatically calculates the percentage of body fat, points awarded per event, total score, final category based on score, and age group.
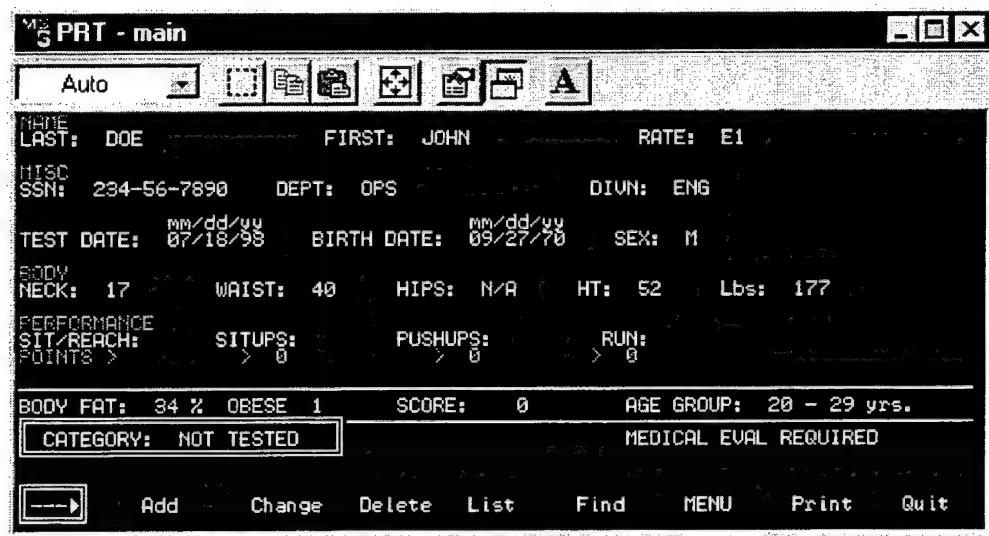
Figure 4. Record Information Screen

To prevent duplication of effort, the program provides the ability to fabricate a new database for each exam cycle by creating a template of the most recent database. Derived from this feature, the program tracks consecutive PRT failures, medical waivers, overfat result, and obese diagnoses (9 maximum each for a total of 4.5 years of data).[1]

Finally, the program reduced administrative demands by making available a selection of sample forms. The sample forms are selected from a menu such as the one depicted in Figure 5. It includes a list of Outstanding PRT results, a Letter of Achievement for PRT Improvement and body fat improvement, a list for those falling in the overfat or obese categories, and a Page 13 for failure of PRT.

---

[1] The obese and overfat categories are no longer used.
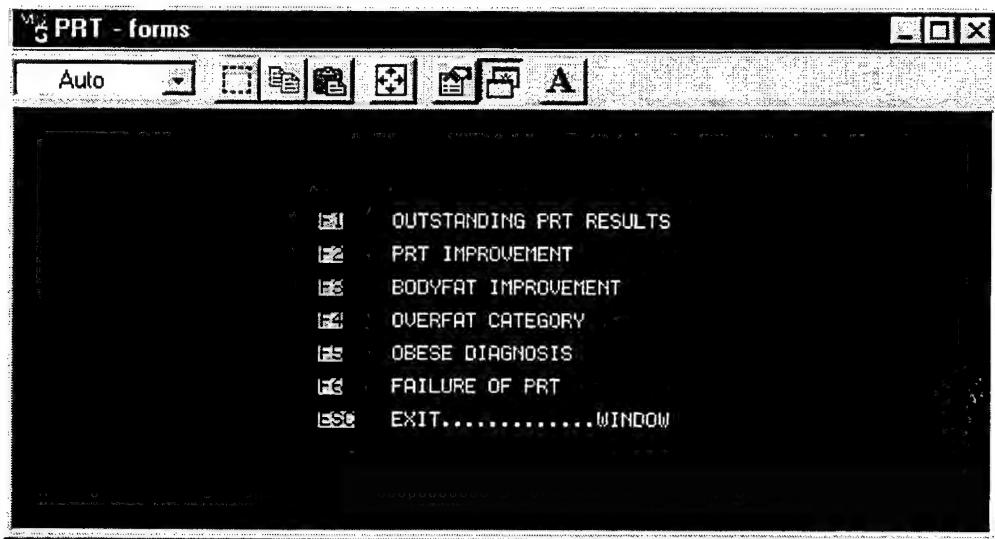
Figure 5. Forms Screen

4. Limitations

The program although very useful has several limitations. Records in the database are maintained by name. If a record with the exact name entered already exists within the PRT database, the newer record will not be entered into the database unless the SSN is different from the existing record. However, it is possible to save two records with the same SSN and different names. It is also possible to save records with blank fields, including critical fields such as name and SSN. Fields which are needed for computation of measurements and scores, such as sex, birth date, and individual event scores may be left blank inhibiting the automatic computation of other fields or placing "UNKNOWN" in that field, such as age group and PRT category. Furthermore, it is possible to enter score information for a member who is said to be "Not tested" due to body fat since the fields are not blocked.

The program was designed for yesterday's technology. Faster computers with large amounts of cheap memory and generally running Windows applications have

replaced the 286 computer. Moreover, IT-21 standards for personal computers are at a minimum [Ref. 6]:

1. 200 MHz Pentium Processor CPU

2. 64 MB EDO RAM

3. 3.0 GB Hard Drive

4. 3.5 inch floppy disk drive

5. 8X IDE CD ROM

6. Dual PCMCIA/PC Card Reader

7. PCI Video with 2MB RAM

8. 17 inch monitor (1280-1024)

9. Pointing devise (trackball or mouse) and keyboard

10. SoundBlaster (compatible) audio card with speakers

11. CPU compatible 100 Mbs fast Ethernet NIC

Finally and perhaps the most important limitation, this program is based on OPNAVINST 6110. Since then, there have been four revisions to the policy and guidance for the administration of the PRT. For example, the obese and overfat categories are no longer used. The Weight Management Program has replaced the Level 2 and Level 3 program. The sit-and-reach is now Pass or Not Pass. Failing this event does not constitute failing the PRT but requires participation in the Command Conditioning Program. Prior to the current instruction, all members had to be measured regardless of weight and height. This prerequisite has been changed and members are now required to be measured only when the weight and height standards are not met.

In addition, the program does not comply with the latest standards. OPNAVINST 6110.1E changed the height and weight standards to accommodate taller females (72 to 80 inches tall), taller men (77 to 80 inches), and shorter men (58 to 61 inches). Furthermore, the maximum body fat for women has been raised from 30 to 33. Scoring computations for women were revised as well.

B. ALPHA 4 DOS VERSION

In 1996, AOC(AW) Michael A. Gissel, NAS South Weymouth, developed a relational database management program using Alpha 4, Version 6. Although a DOS program, the program was very thorough. Serving as a CFC, he saw the need to develop a program that would be user friendly, versatile, and able to provide a wide range of statistical data and reports. The program would eliminate the administrative burden on CFC's, provide rapid individual or command status, and store an individual's fitness history for at least four years. [Ref. 7]

1. Hardware and Software Requirements

The system requirements called for an IBM compatible personal computer with at least 2MB of XMS memory, a 3.5-inch drive and a hard drive, running on DOS 3.1 or higher. It also required a 386 or higher CPU with at least 8MB of RAM. In addition, input devices can be either the keyboard or mouse. For printing forms and reports, a laser printer capable of generating reports in both landscape and portrait format was also necessary. [Ref. 8]

The user installs the program from three floppy disks using a supplied installation utility. The setup program uses 4.80MB of memory. Six folders are created for each linking database, labeled Bodyfat, Category, Members, Run, Swim and Weight. To start

14

the program, the user enters A4V6RT.BAT at the DOS prompt, or creates a shortcut in a Windows environment.

Although a major improvement over the 1988 version, the program is not widely distributed. AOC Gissel purchased a license to be able to distribute the program to Naval commands and approximately twelve commands have been given a copy.

2. Program Design

The program is menu driven. Selections are made by highlighting the desired option. The user can navigate the screens with the mouse, the arrow keys or the return key. Figure 6 depicts the Main Menu.

Figure 6. Main Menu

The database consists of individual records. The database name must be selected from a choice of FY1996 through FY2002. Upon selection of the database the record

input screen appears, Figure 7. The record consists of 160 fields. Approximately thirty fields are calculated by the program, such as body fat percentage, age, points, etc. The calculated fields are read only. In addition, there are several link fields. The program generates link numbers to maintain connection with other databases. For example there is a link field, CATLINK1A, for the PRT category and a BFLINK1A for the Body Fat category. The user must enter this generated link number in the field right below it as prompted by the program.

Figure 7. Page 1 of the Record

3. Functionality

The Alpha 4 program is more thorough than the 1988 DOS version. It provides the user with more options and greatly reduces the administrative demands on the CFC. For example, the program provides an extensive selection of forms and reports. Among the forms provided are Award Certificate, Medical Board Letter, Page 13, and a Weigh-

16

In Window Waiver for those who did not take the measurements on time and are within standards but still want to participate in the PRT. It also gives a myriad of reports, such as Annual PRT Readiness Report, Cleared to take PRT, Current Spring PRT Report, Current Fall PRT Report, Advancement Report Spring Cycle, Advancement Report Fall Cycle, PRT Failure Rate Spring Cycle, PRT Failure Rate Fall Cycle, Scoring Sheet and Telephone listing.

A record consists of all the information pertaining to the member for the last four years. This information is presented to the user in two pages, Figure 7 and Figure 8. The first page contains the member's identification information. It also lists the necessary information to allow a member to participate in the PRT. It identifies the active duty status of the member, the current physical information, the 4-year PRT and failure history, and recommendations based on PRT score. In addition, it covers information related to the pink folder. It registers if the physical is current, and if the member answered "yes" to any medical screen's questions. It lists whether the member was measured and if screening is required due to measurement failure and the date the screening was completed. If the member is not cleared, a Page 13 must be issued and the issuing date will be entered in the PG 13 Date field.

Page 2 of the Record, Figure 8, is the actual PRT information. It lists the information for the Spring and Fall cycles of the PRT. The calculated fields on this screen are BODYFAT%, BFRESULT, event categories and points, TOTAL, PRT RESULTS, and EVAL/FITREP CODES. If the member is not cleared to take the PRT, the user will not be able to enter TESTED DATE. In addition, this screen will display a message, regarding the status of the weight standard and PRT clearance of the member.

A4V6RT - ALPHART

Auto   A

LT___ DOE, JOHN   AGE: 32 SEX: M DEPT: ADMIN   SEARCH:

M74220 1995 SPRING CYCLE M27 074 0   M74215 1995 FALL CYCLE M25 074 0

MEASURED: 05/05/1996 BODYFAT* 29   MEASURED: 11/11/1996 BODYFAT* 28
HEIGHT: 74.0 WEIGHT: 220 BFRESULT: FAIL   HEIGHT: 74.0 WEIGHT: 216 BFRESULT: FAIL
NECK: 17.0 WAIST: 44.0 HIPS: 0.0   NECK: 15.0 WAIST: 40.0 HIPS: 0.0
         NOT WITHIN STANDARD   NOT WITHIN STANDARD

MEMBER CLEARED AND TESTED: 11/11/1996   MEMBER CLEARED AND TESTED:
       SCORE   PTS CAT   SCORE   PTS   CAT   M32 0
SR(SR)   P   0   P   SR(SR)   0
CU(C)   55   55   CU(C)   0   0
PU(P)   66   99   WAIVER   PU(P)   0   0   WAIVER
RUN(R) 1111   81   CODE: RUN(R)   0   0   CODE:
SWIM(S) 0   0   SWIM(S) 0   0
     TOTAL: 235   TOTAL: 0
PRT STATUS: EXC   PRT STATUS:
PRT RESULTS: PASS   PRT RESULTS: N/A
EVAL/FITREP CODE: P/NS   EVAL/FITREP CODE: /NS
PAGE 13 ISSUED: 05/05/1996   PAGE 13 ISSUED:
PAGE 13 SIGNED BY MEMBER: 05/05/1996   PAGE 13 SIGNED BY MEMBER:
CHANGE   Record : 1   ↑ Page: 2
PRESS (F3) FOR OFFICER RANK SELECTIONS ONLY.
F2Options F10Save EscForget F3Choices

Figure 8. Page 2 of the Record

Finally, the Alpha 4 program provides additional functionality, which the 1988 DOS version does not provide. The program allows the ability to merge from other files, such as Lotus 1-2-3, Dbase, VisiCalc, ASCII File, PFS File, and WordPerfect. It also gives the user the option to create forms, index database by department, name or number, global delete/update of records, selection of printer, selection of screen colors, and append, backup or erase a database.

4. Limitations

Although a major improvement over the 1988 DOS version, the program has some of the same limitations of the 1988 program. The Alpha 4 program does not comply with OPNAVINST 6110.1E and does not take advantage of the requirements set by IT-21.

Furthermore, the implementation is very crude. This program requires the user to be familiar with relational databases. The User Manual gives a limited explanation on what a relational database is and how to use it. It emphasizes the need to fill out all of the fields in the record. However, the fields in the record may be left blank with the exception of the link fields. Leaving the fields blank prevents the calculation of other fields. In addition, you can enter data in the fields other than the options listed in the user's manual, and there is no validation of dates entered. For example, the user is able to enter a date of measurement past the date of the PRT.

As part of this relational database, the program gives the user the option to query through the QUERY menu. Queries must be entered with relational algebra. Most CFCs will not be familiar with the specification of relational algebra to construct a complicated query to arrive at the desired answer. To formulate a query, the program requires the user to specify the arguments by programming field name, and the screen that is generated by the QUERY command gives an alternative of all 160 fields, including the link fields.

The menus, options and fields are not self-explanatory. Since the User Manual does not address every option or menu, it is difficult to determine what is available or what action to take in some cases. For example, at times a Password screen appears with no explanation on how to escape and no way to cancel without restarting the program. In other cases, options are listed which are of no use to the user. For example, a third index for the database, which places all records in order by record number, is available but the User Manual recommends not using it

Finally, there are two programming shortcomings. The available databases are for FY1996 – FY2002. After that year, there is no instruction on how to add databases.

Secondly, the link fields, although necessary for the performance of the program, require additional work by the user. They must be entered exactly as generated by the program. Otherwise, an error message is displayed. A practical application should reduce the amount of keystrokes while maintaining effectiveness. Furthermore, the program does not provide a method for importing data from the 1988 DOS version used by the majority of commands.

III. DESIGN

A. CONCEPTUAL SCHEMA

A conceptual schema is a concise description of the data requirements of the users and includes detailed descriptions of data types, relationships, and constraints [Ref. 9]. To provide a clearer description of the requirements, the conceptual schema is displayed by means of the graphical notation in Figures 9 through 14, the Entity Relationship diagrams. In designing the PRT database, the following requirements were considered:

1. The command is organized into departments. Each department has a unique name. To ease conversion to a centralize database the command entity was retained in the Entity Relationship Model, Figure 9. The command has a unique name, an address, a Commanding Officer, a Point of Contact, a Type Commander, multiple Unit Identifier Codes (UIC) and a Defense Switched Network and Commercial phone numbers.

2. The command administers two PRTs per year. The date and a unique PRT Number are maintained for each PRT.

3. Every servicemember would participate in the PRT and his information such as name, SSN, date of birth, paygrade, sex, telephone, day checked aboard, day checked out, active duty status, four-year PRT history, current physical, and participation in Bureau of Medicine Weight Management Program (WMP), would be recorded in the database.

4. The results of every PRT for every servicemember would be maintained by the database. This information would include reference to completion of Section B of the pink folder, measurements, event scores, waiver codes, Page 13 information and comments if necessary.

21

B. ENTITY RELATIONSHIP DIAGRAM

The Entity Relationship (ER) diagram describes data as entities, relationships, and attributes [Ref. 9:p. 42]. Figures 9 depicts the overall ER model. In this model, there are five entities, three of them being weak entities identified by a double rectangle. Weak entities relate to their identifying owner through the identifying relationship. In Figure 9, Department and PRT are weak entities of the Command entity. Results is a weak entity having as identifying owner Servicemember. The figure also shows cardinality ratios and participation constraints. Participation constraint specifies whether the existence of an entity depends on its being related to another entity via some relationship type [Ref. 9:p.54]. These can be total participation, a double line, and partial participation, a single line between the relationships and the entities. The cardinality ratio is displayed by using 1, M, and N. For example, there are several servicemembers working for one department, thus a 1:N relationship.
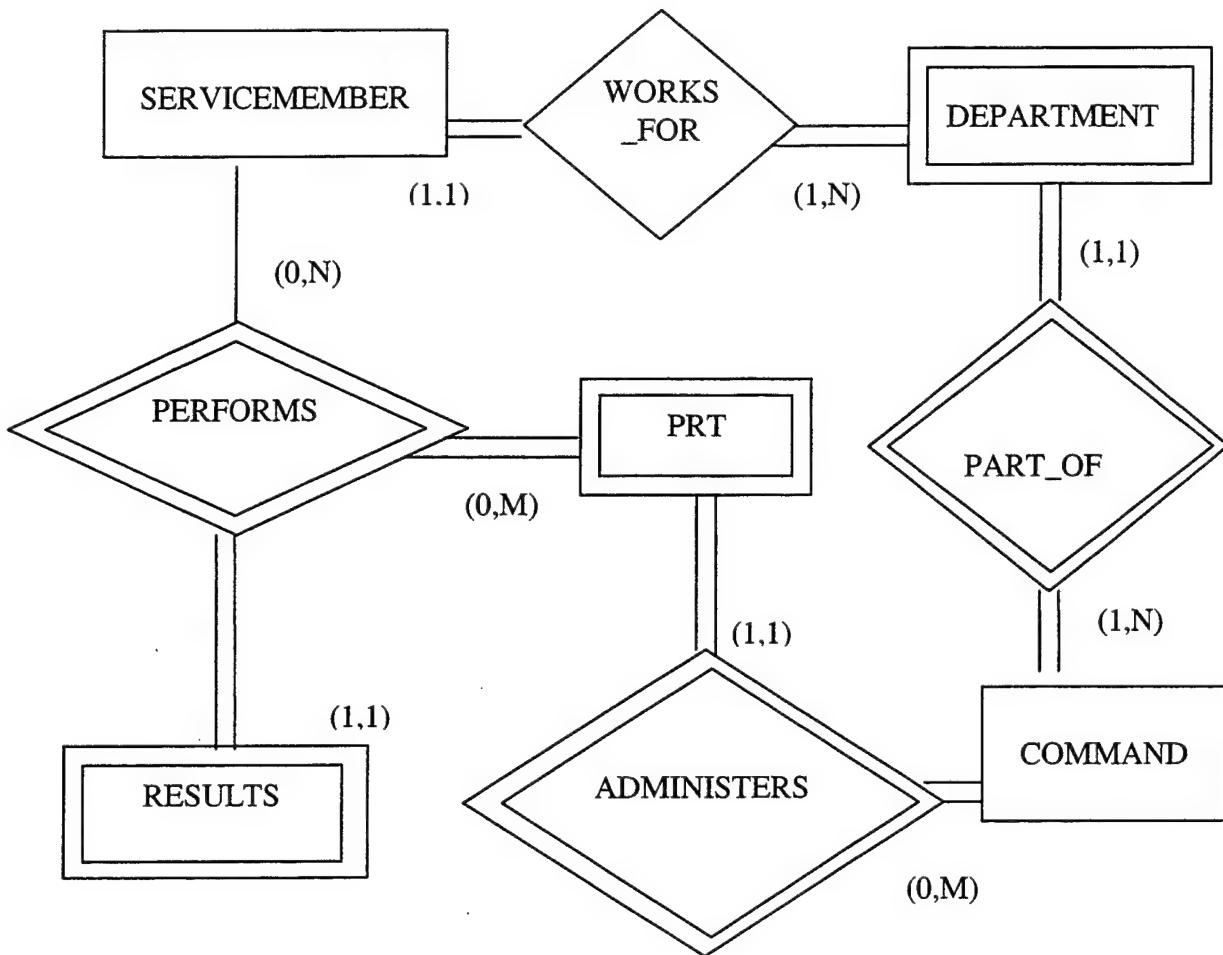
Figure 9. Entity Relationship Diagram

Figures 10 through 14 show the attributes for each entity. The Command entity, depicted in Figure 10, consists of six simple attributes: command name, commanding officer's name, address, point of contact (POC), and type commander (TYPECOM). The UIC is a multivalued attribute since one command can have several UICs. The phone number is a composite attribute consisting of the DSN and the commercial phone number. Envisioning a centralized database, the command entity was included in the relationship diagram.

23
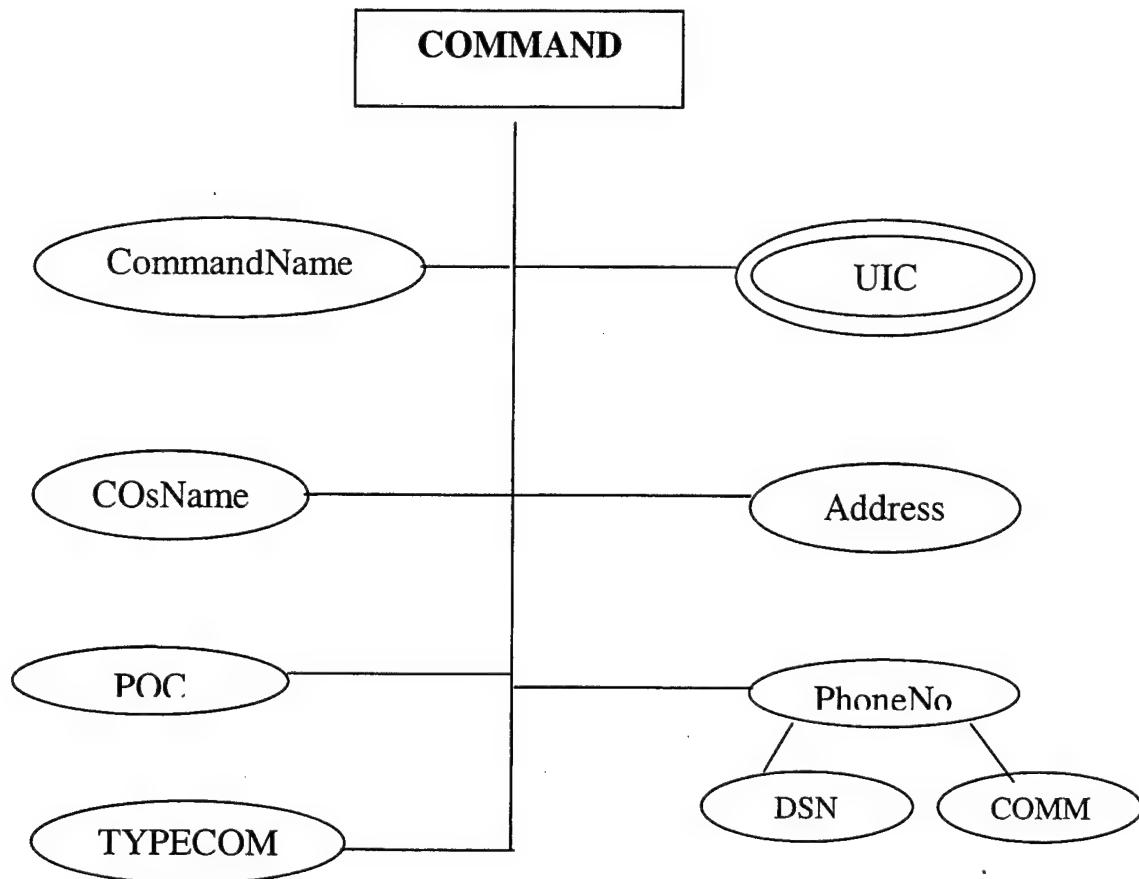
Figure 10. Command Entity and Attributes

Figure 11 depicts the Department entity and its attributes. The Department entity

consists of one stored attribute and nine derived attributes. The stored attribute is a

unique name for each department at a command. The derived attributes are identified by

a dotted line. These derived attributes are used for statistical purposes only and were

added to follow the model of Chief Gissel's program.
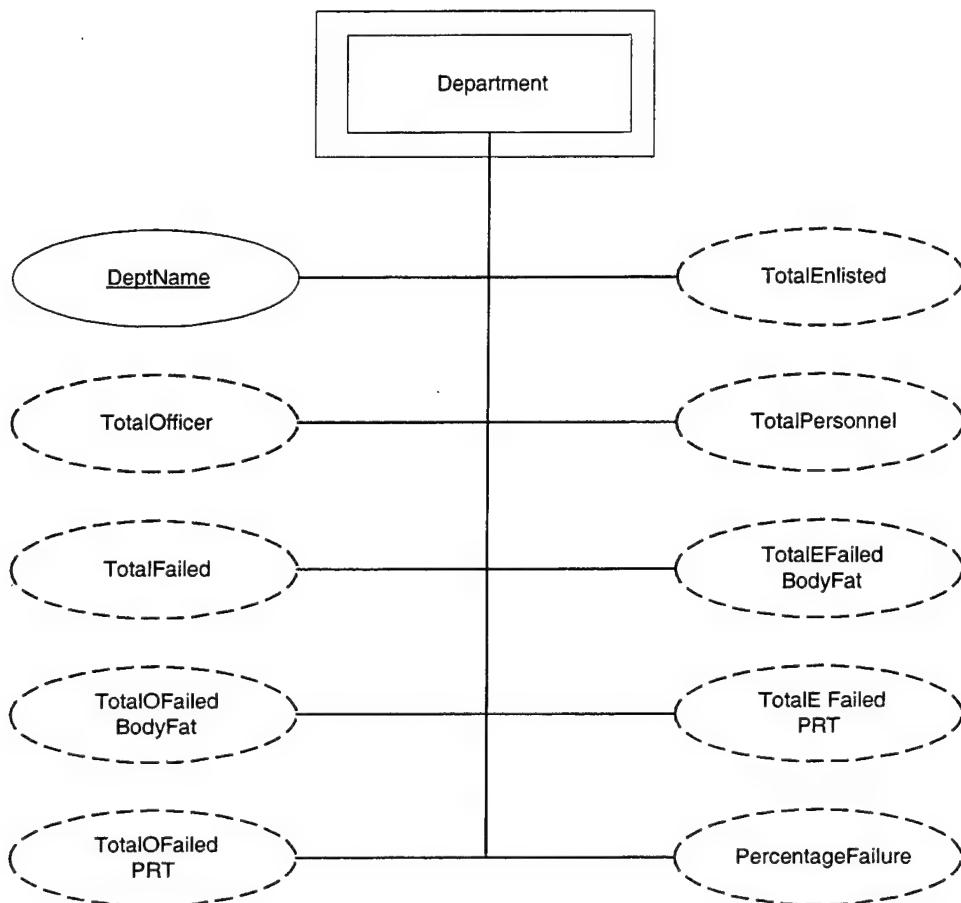
Figure 11. Department Entity and Attributes

The Servicemember entity, Figure 12, consists of 23 attributes, 12 stored and 11 derived. The primary key for this entity is the Social Security Number (SSN). The only composite attribute is the name, consisting of last name, first name and middle initial. Among the stored attributes are the name, SSN, paygrade, active duty status, sex, date of birth, telephone extension, date checked aboard, date checked out, current physical, date WMP scheduled and date completed. The derived attributes consist of the age, total PRT failures in four years, when physical expires, WMP aftercare completion date, and recommendations for enlisted personnel based on PRT failures.
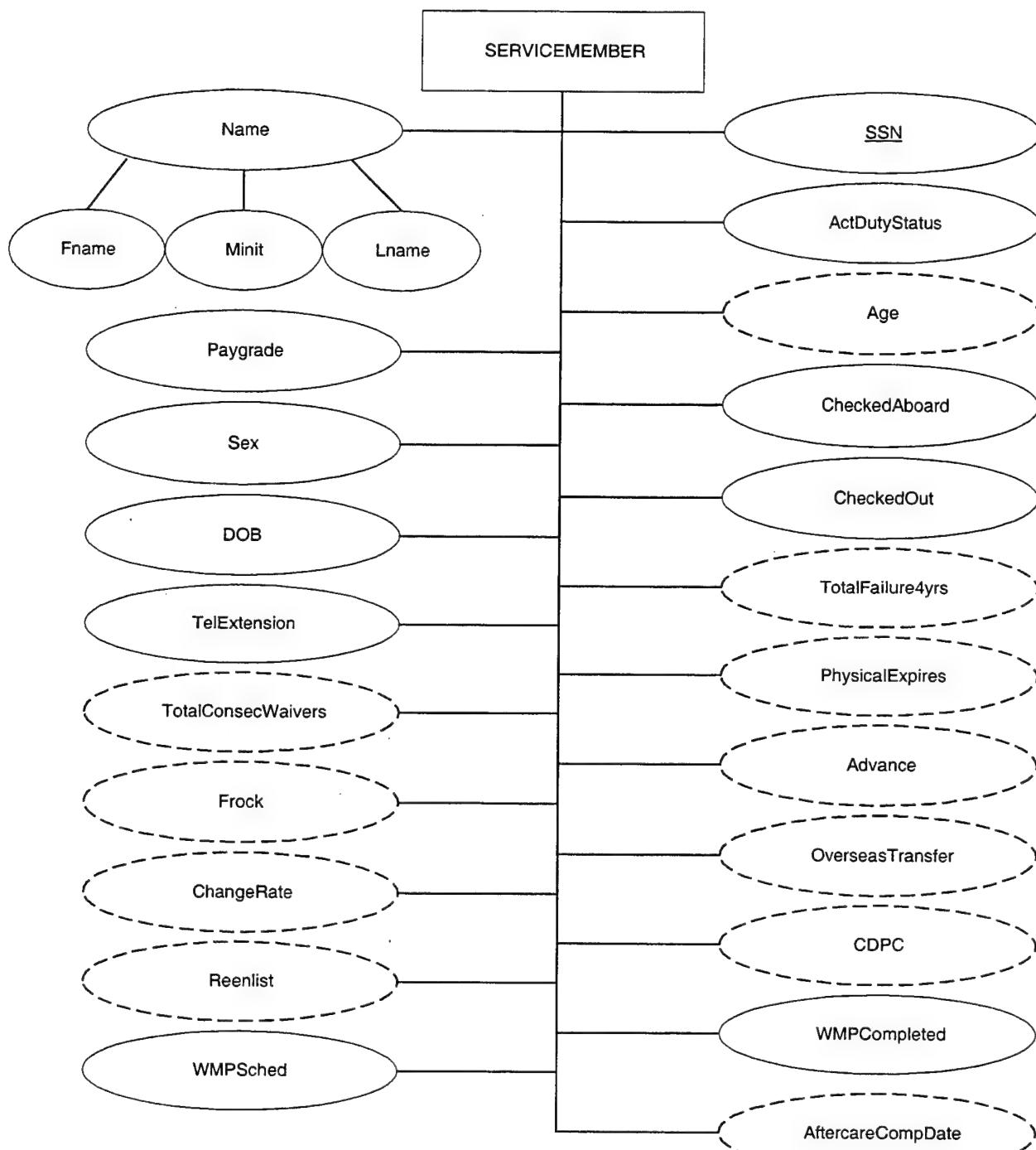
Figure 12. Servicemember Entity and Attributes

Figure 13 depicts the Results entity and its attributes. This entity consists of 33 attributes, 17 stored and 16 derived. The stored attributes include answering "yes" to Section B of the pink folder, Section B completion date, screening date, height, weight, neck measurement, waist measurement, hip measurement[2], waiver code, sit and reach score, number of curl ups and push ups, run or swim time, swim type, date page 13 issued and signed, comments and measurements date. Among the derived attribute are body fat percentage, body fat code result, whether based on Section B screening is required or member is cleared, curl ups points, curl ups category, push ups points, push ups category, run/swim points, run/swim category, total points, total category, and evaluation/fitness report codes.

---

[2] Women only.
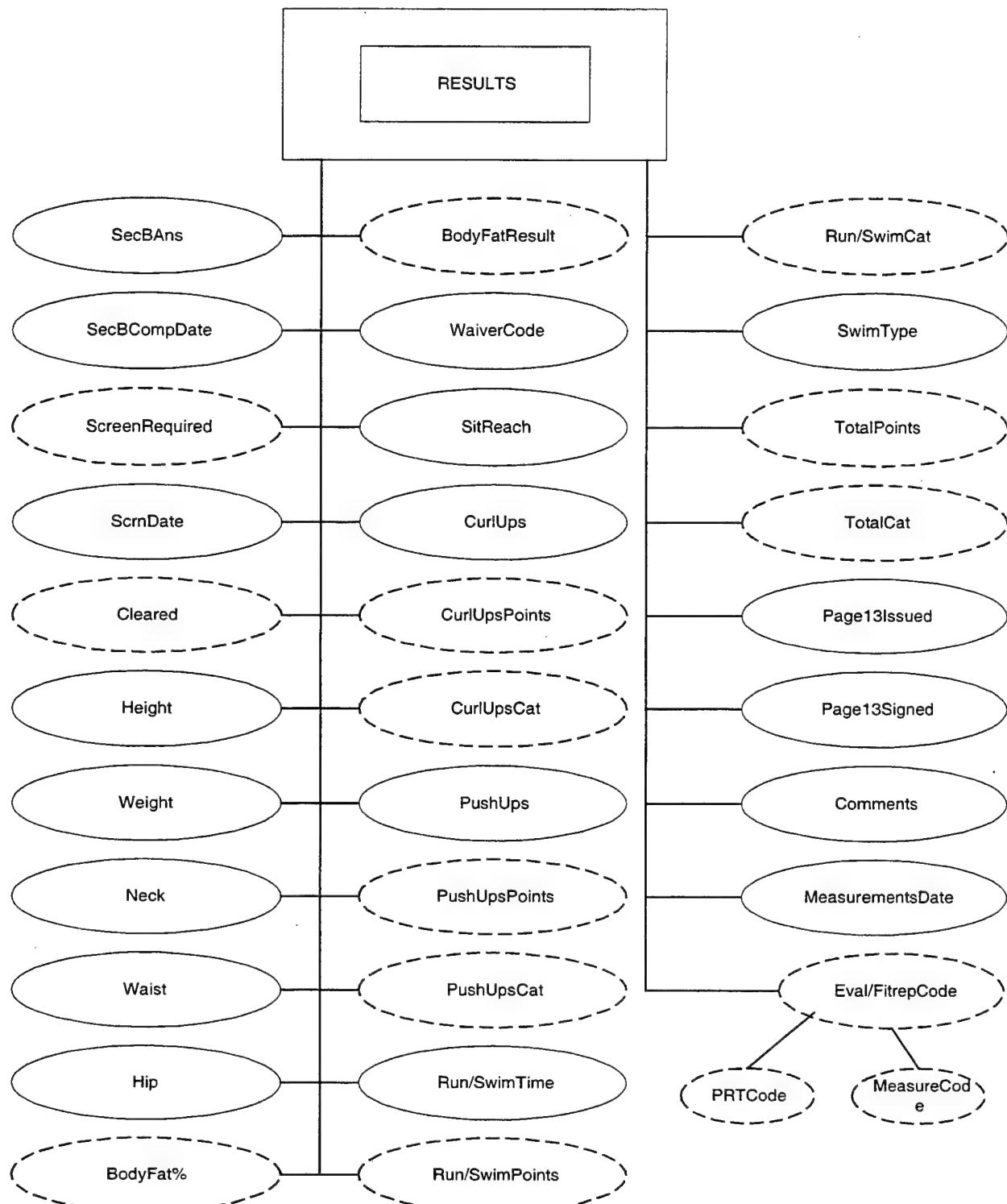
Figure 13. Results Entity and Attributes

The last entity in the ER diagram is the PRT, Figure 14. The PRT has 13 attributes, two stored and eleven derived. The derived attributes are necessary to compile the Command Summary. PRT number is the partial key of this weak entity.
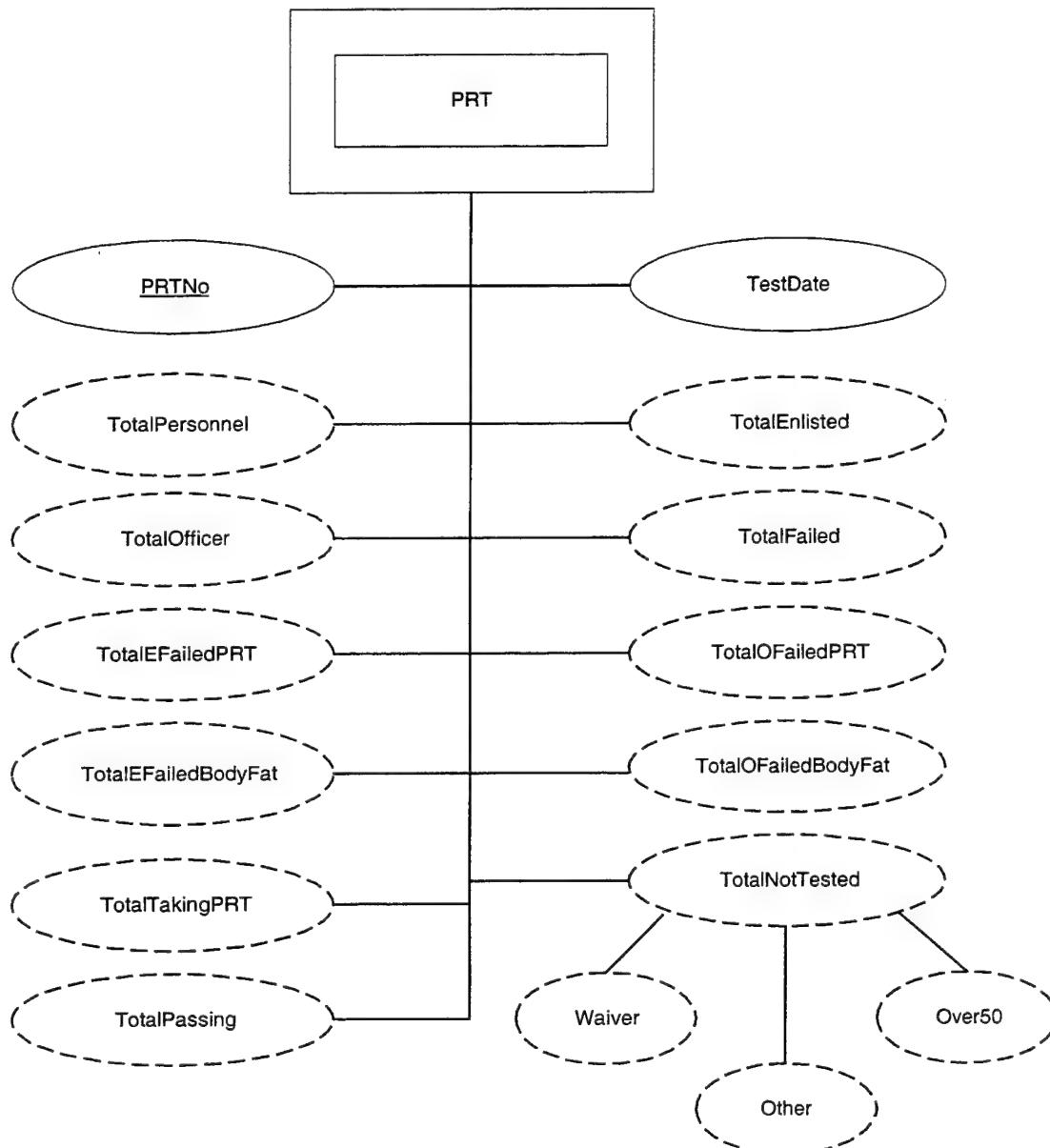
Figure 14. PRT Entity and Attributes

C. DATABASE IMPLEMENTATION

The PRT database was implemented using Microsoft Access. Access 97 was chosen based on the IT-21 requirements and its compatibility with Visual Basic 5.0. Microsoft has provided a software system that manages how data is stored and retrieved through its database engine, Microsoft Jet 3.5. This database engine ships with both Microsoft Access 97 and Visual Basic 5.0.

1. Access 97 DBMS

The first step in designing the PRT database was to construct the tables. Each entity became a table with attributes listed as fields' descriptions. These tables comprise the primary tables. A total of 6 primary tables were created. Only their stored attributes were created as fields. The fields' properties such as data types, length, validation rules, and others were set. The next step was to define relationships among the primary tables. Microsoft Access enables the user to define the relationships among tables. When the user defines a relationship, the database engine can enforce referential integrity among the related tables.[Ref. 10:p.37] By enforcing referential integrity on the tables, the database engine prevents the user from removing a record when there are other records related to it in the database.

Figure 15 depicts the primary tables and their relationships.
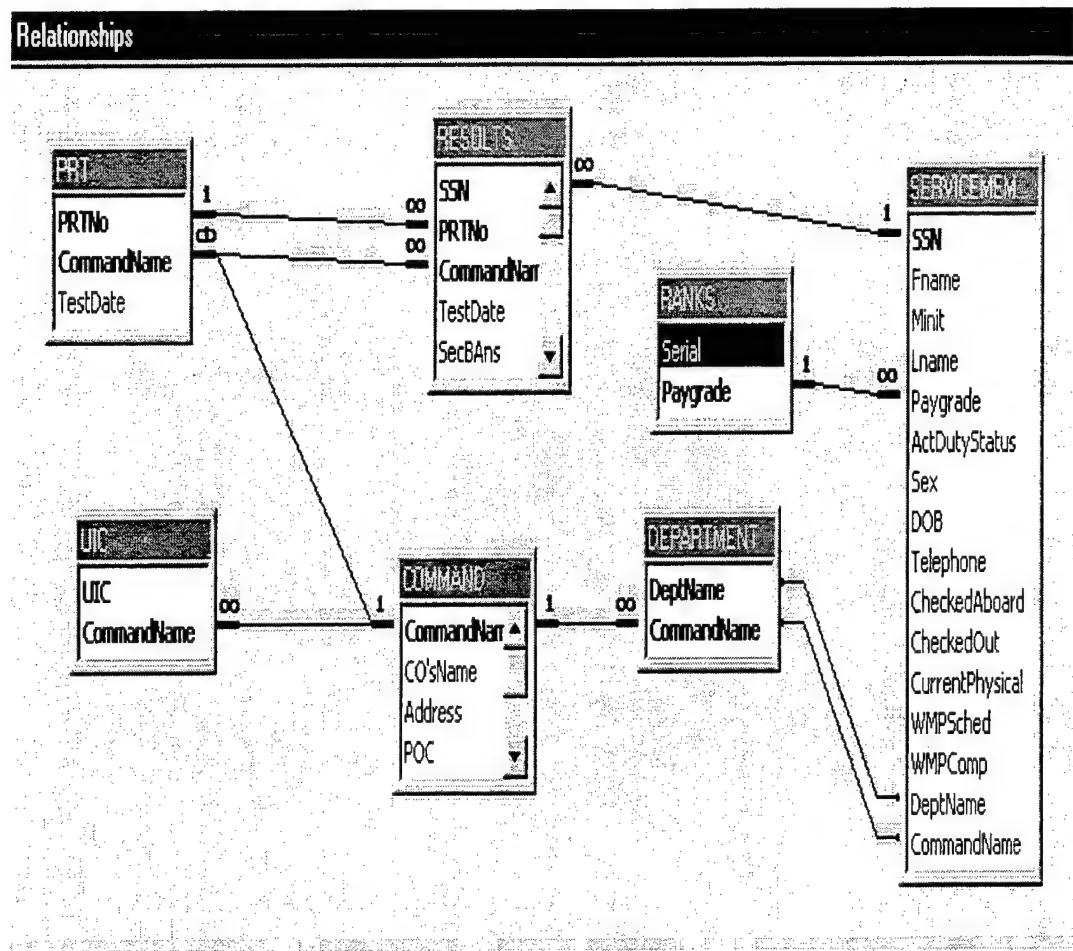
Figure 15. Relationships Diagram

There were other administrative tables included in the database. These five tables included CurlUpsCat, PushUpsCat, TestCategory, TestRun/Swim and Waiver Code. The first four tables were used to determine the individual event category based on scores achieved. The data for these tables was taken from Appendix A to OPNAVINST 6110.1E. The last table was used to list the waiver codes. These waiver codes were designed to enable the computation for the Command Summary. The waiver codes are assigned in the Results table and refer to reasons why the member did not participate in an event or the entire PRT. A default value of zero was assigned to waiver code in the

Results Table. If no other waiver code is entered, the member is considered to have participated in all the events of the PRT. The Waiver Codes were assigned as follows:

Waiver Code	Waived Reason
1	Sit-And-Reach
2	Curl Ups
3	Push Ups
4	Run
5	Swim
6	Curl Ups and Push Ups
7	Curl Ups and Run (Swim)
8	Push Ups and Run (Swim)
9	All (Medical)
10	All (Over 50)
11	All (Other)

A total of 45 queries were created. With the exception of one delete query, DeleteOldResults, the remainder were select queries.   In the following list the name of the queries is followed by a list of queries or tables used to derive the information.

BodyFat (Results,Circumference)

Circumference (Servicemember, Results)

CurrentRecommendations (4YrFailure, Results)

DeleteOldResults (Results)

EnlistedResults_Test (FindAgeCategory, CurlUpsCat, PushUpsCat,

TestRun/Swim, TestCategory)

FailedByDept (Servicemember, TestCategory, Results, Ranks)

LatestBodyFat (BodyFat, LatestResults)

LatestPRT (PRT)

LatestPRTNo (LatestPRT, PRT)

LatestResults (Results, LatestPRTNo)

LON (Results, LatestBodyFat, OfficerWhoFailed, Scores)

OfficerResults_Test (FindAgeCategory, CurlUpsCat, PushUpsCat,

TestRun/Swim, TestCategory)

OfficersWhoFailed(LatestPRTNo, WhoFailed, Servicemember)

Oustanding (LatestPRTNo, Scores)

PassFail (Scores)

RecPRT (Results, PassFail, LatestBodyFat)

Scores (FindAgeCategory, CurlUpsCat, PushUpsCat, TestRun/Swim,

Test Category)

TotalEFailedBodyFat(LatestBodyFat,Servicemember,Ranks)

TotalEFailedPRT(LatestPRTNo, WhoFailed)

TotalEFailingCurlUps(LatestPRTNo, Scores)

TotalEFailingPushUps(LatestPRTNo, Scores)

TotalEFailingRunSwim(LatestPRTNo, Scores)

TotalEFailingSitReach(Scores, Latest Results)

TotalEnlisted (Ranks, Servicemember, LatestResults)

TotalENotTestedOther (Servicemember, Results, Ranks, LatestPRTNo)

TotalENotTestedOver50 (Servicemember, Results, Ranks, LatestPRTNo)

TotalEPassingPRT (Servicemember, PassFail, Ranks, LatestPRTNo)

TotalETakingPRT (Servicemember, Results, Ranks, LatestPRTNo)

TotalFailure4Yrs (PassFail)

TotalMedicalWaiverEnlisted (Servicemember, Results, Ranks, LatestPRTNo)

TotalMedicalWaiverOfficer (Servicemember, Results, Ranks, LatestPRTNo)

TotalOFailedBodyFat (Servicemember, Ranks, LatestBodyFat)

TotalOFailedPRT (WhoFailed, LatestPRTNo)

TotalOFailingCurlUps (Scores, LatestPRTNo)

TotalOFailingPushUps (Scores, LatestPRTNo)

TotalOFailingRun/Swim (Scores, LatestPRTNo)

TotalOFailingSitReach (Ranks, Servicemember, LatestsResults)

TotalOfficer (Ranks, Servicemember, LatestResults)

TotalONotTestedOther (Servicemember, Results, Ranks, LatestPRTNo)

TotalONotTestedOver50 (Servicemember, Results, Ranks, LatestPRTNo)

TotalOPassing (Servicemember, PassFail, Ranks, LatestPRTNo)

TotalOTakingPRT (Servicemember, Results, Ranks, LatestPRTNo)

WhoFailed (Scores)

Twenty-four of the above queries were used for the Command Summary. The Outstanding and LON queries were designed to be used with Crystal Reports 6.0 for Award Certificates and Letters of Notification. The RecPRT and CurrentRecommendations were also designed to be used with Crystal Reports 4.6 for a listing of recommendations based on the current PRT and 4-year history. These

recommendations will determine if the member should be recommended for frocking, reenlistment, advancement, overseas transfer, changing rate and Command Directed Conditioning Program.

Two queries required formulas. The Circumference query was designed using the formula listed in OPNAVINST 6110.1e. The circumference formulas are different for men and women. This formula takes into account the neck, waist, and hip (women only) measurements. These measurements will be input into the Results table whenever the member fails the height and weight standards. The BodyFat query which determines the percentage of body fat based on the member's circumference and height was designed using Dr. J. Hodgdon's formulas. [Ref. 11] The formulas are as follows:

Women – 495/(1.29579-.35004(log(circumference))+.22100(log(height)))-450

Men – 495/(1.0324 - .19077(log(circumference))+.15456(log(height)))-450

There were three queries designed to fill a data grid in the application using a filter. These were the EnlistedResults_Test, OfficerResults_Test, and FailedByDept. The remaining queries were designed to be used in other queries.

2. Visual Basic 5.0

The application was designed using Visual Basic 5.0. There were 19 Forms in the project, one module and six Crystal Reports 6.0 designers. Thirteen forms were individually designed to interact as depicted in Figure 16. The remaining forms were automatically generated to view six reports created with Crystal Reports 6.0.
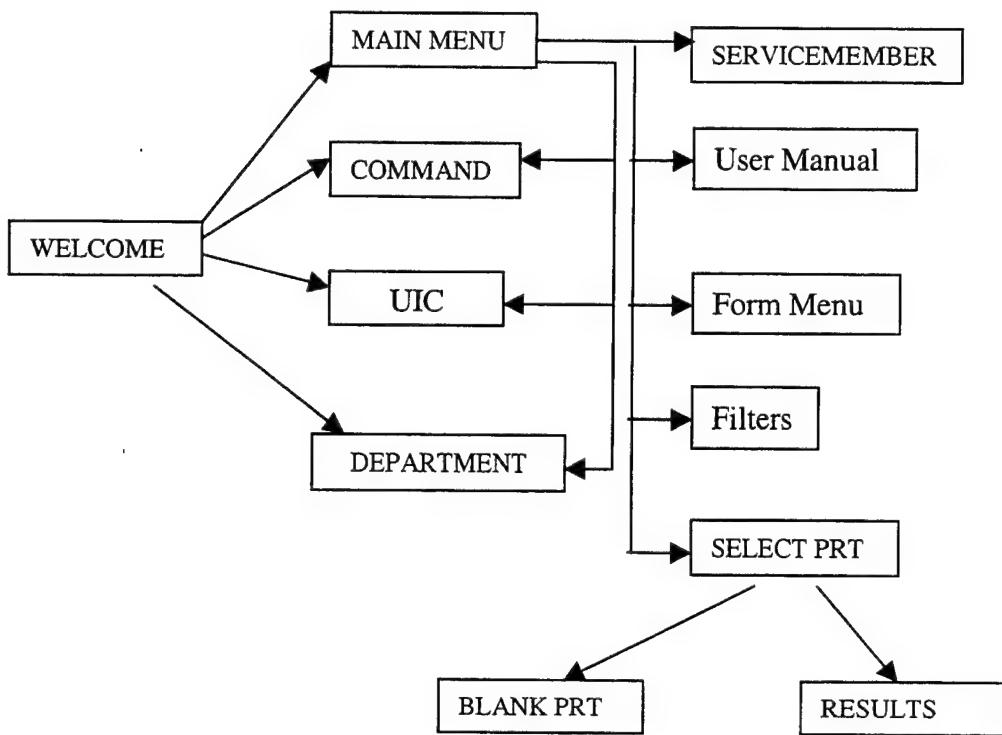
Figure 16. Forms Interaction

Visual Basic 5.0 provides three categories of controls: intrinsic controls, ActiveX controls and insertable objects. Intrinsic controls, such as the command button and frame controls are contained within a Visual Basic .exe file. ActiveX controls, which exist as separate files with a .ocx file name extension include the data-bound grid, combo box, list controls, and so on. Insertable objects are objects such as a Microsoft Excel Worksheet or Microsoft Project Calendar.[Ref.12:p.225]

Only intrinsic and ActiveX controls were used in designing the forms for this application. Most of the functionality of the program is achieved by selecting buttons, entering text in text boxes and displaying data in labels. The ActiveX components were used for opening the User's Manual with a Common Dialog Box and for filtering recordsets in the Filters form with a Data-bound grid. In addition, reports created with

Crystal Reports 4.6 require a Crystal Reports ActiveX component to be included in the opening form.

Microsoft Jet Database Engine allows you to access the database in several ways. The database can be accessed through a Data Control, Data Access Objects or Structured Query Language (SQL). In this project, all three forms of access were used. The Data Control manages the connection between a Visual Basic form and a database [Ref. 10: p. 47]. In addition, data controls are the simplest way of gaining access to databases in Visual Basic, whether the files are in Visual Basic's native format (shared with that of Microsoft Access), in an external format such as that of dBASE, or in a client/server database environment. [Ref. 10:p. 47] For this reason, data controls were used in most of the forms to access the database.

Data Access Objects are most commonly used to manipulate data in an existing database [Ref. 10:p. 139]. In particular, they were used to retrieve the information for each of the fields in the Command Summary report. They were also used to limit the number of data controls in a form. Furthermore, DAO provides an Execute method that was used to execute the delete query against the database when the user presses the Continue button in the Welcome form. The use of SQL was minimal.

Since data controls were primarily used in the design of this prototype, data-aware controls were available. A data-aware control is any control that has a DataSource property. The DataSource property refers to a Data control because this property connects the user-interface control to the Data control (which in turn connects, or "binds" the user interface to the database.) [Ref. 10: p. 51] Among the data-aware controls used were the DBCombo, DBGrid, DBList, Label, MaskedEdit, and TextBox.

The first form in the application is the Welcome screen, depicted in Figure 17. The Welcome screen displays two buttons, Continue and Exit. If the user has not previously initialized the database, by pressing the Continue button, will display the Command, UIC, and Department screens. Once the user enters the required information in the above forms, the Main Menu will be displayed. If the user has previously initialized the database the delete query will be executed and the Main Menu screen will be displayed. By selecting the Exit button, the user will exit the program.
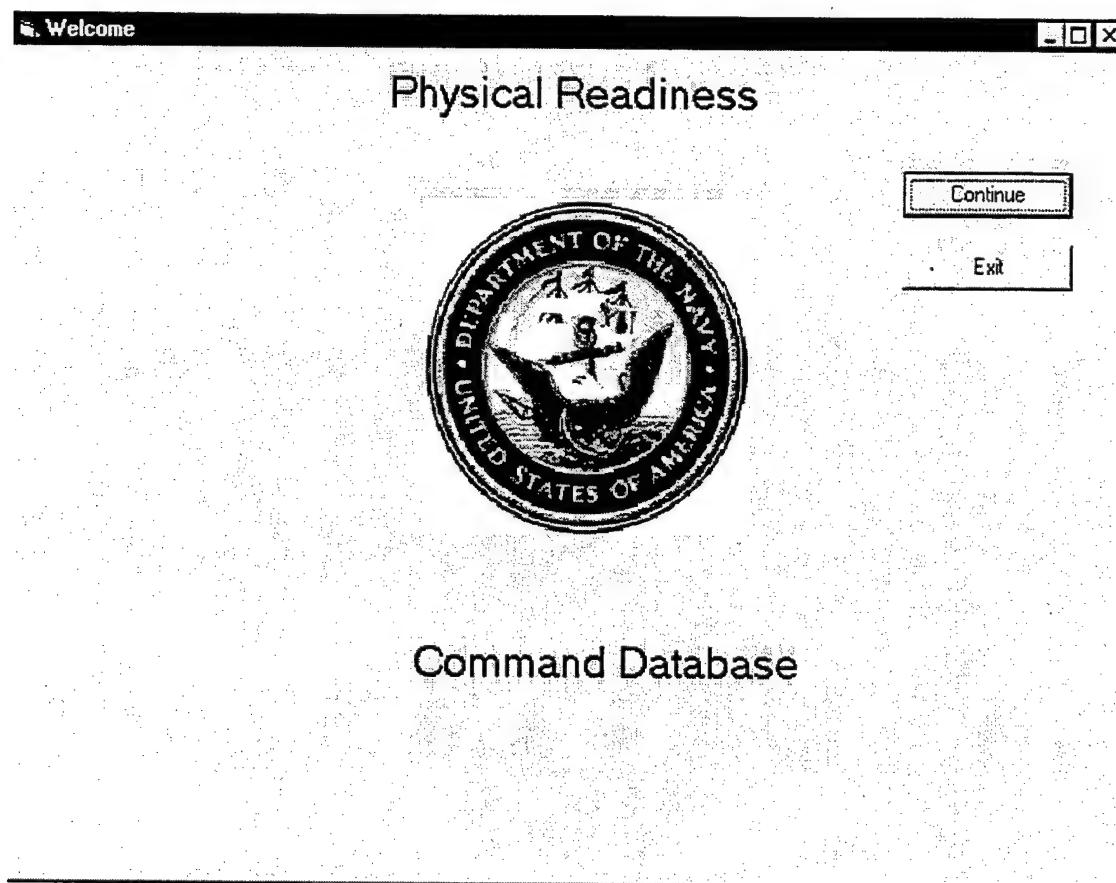
Figure 17. Welcome Screen

The Command screen is shown in Figure 18. Since the database was designed for future centralization, it is possible to add several commands to the database. However,

since the prototype was designed for a single command, the New button is only visible upon initialization of the database. If the user desires to make changes to previously entered information, he can use the Update button. The data control allows the user to scroll through the records in the database.

| Command Information | | |

Command   COMTRAPAC

CO's Name: RADM Roe

Address: 1234

POC: LT Quinones

TYPECOM: CICN

DSN: 234-5567

COMM: (234)456-7789

Add information, press Update.

|◄ ◄ Record   ► ►| Update

Figure 18. Command Information Screen

The UIC screen is shown in Figure 19. The New button allows the user to enter multiple Unit Identification Codes (UICs). However, in designing the Command Summary, the first UIC in the database will be the one displayed. The Update button allows the user to modify information in an

existing record. The Delete button allows the user to delete a record. The data control allows the user to scroll through the records in the database.
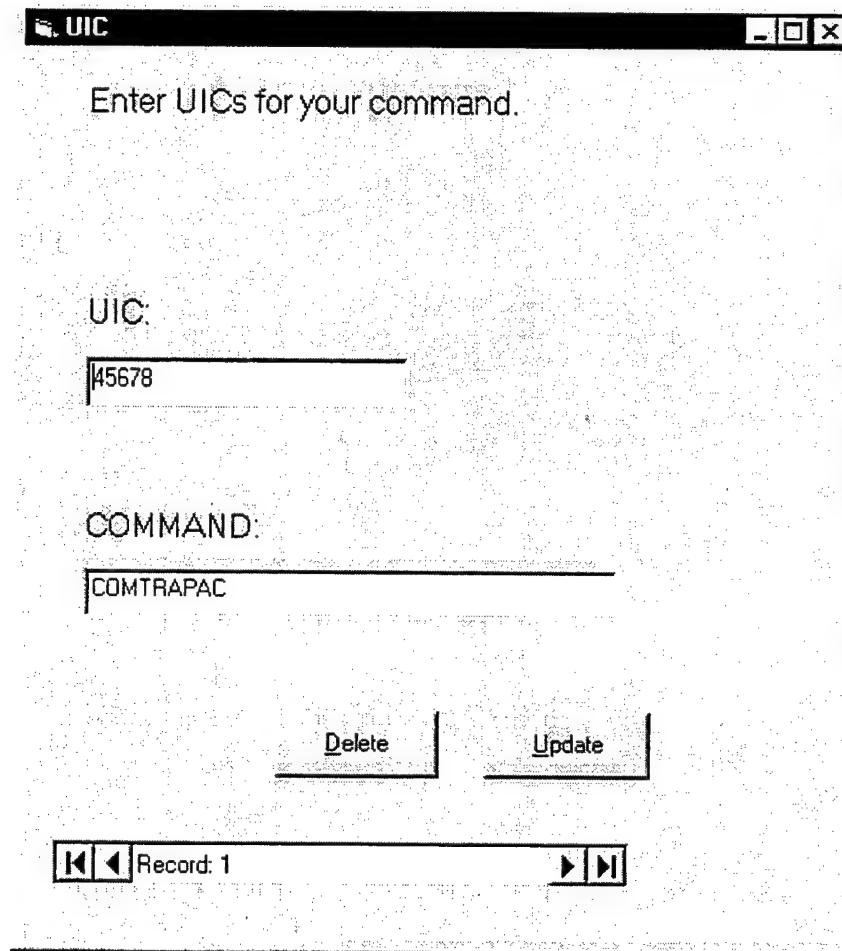
Figure 19. UIC Screen

The Department screen allows the user to input the name of the Departments in his command, Figure 20. The New button allows the user to enter names of all the departments. The Update button allows the user to modify information in an existing record. The Delete button allows the user to delete a record. The data control at the bottom of the screen allows the user to scroll through the records in the database.

DEPARTMENT

Department Name:

ADMIN

Command Name:

COMTRAPAC

Press New, enter Name, press first right arrow in scroll bar below.

[New]   [Delete]   [Update]

|◀ ◀ Record: 1   ▶ ▶|
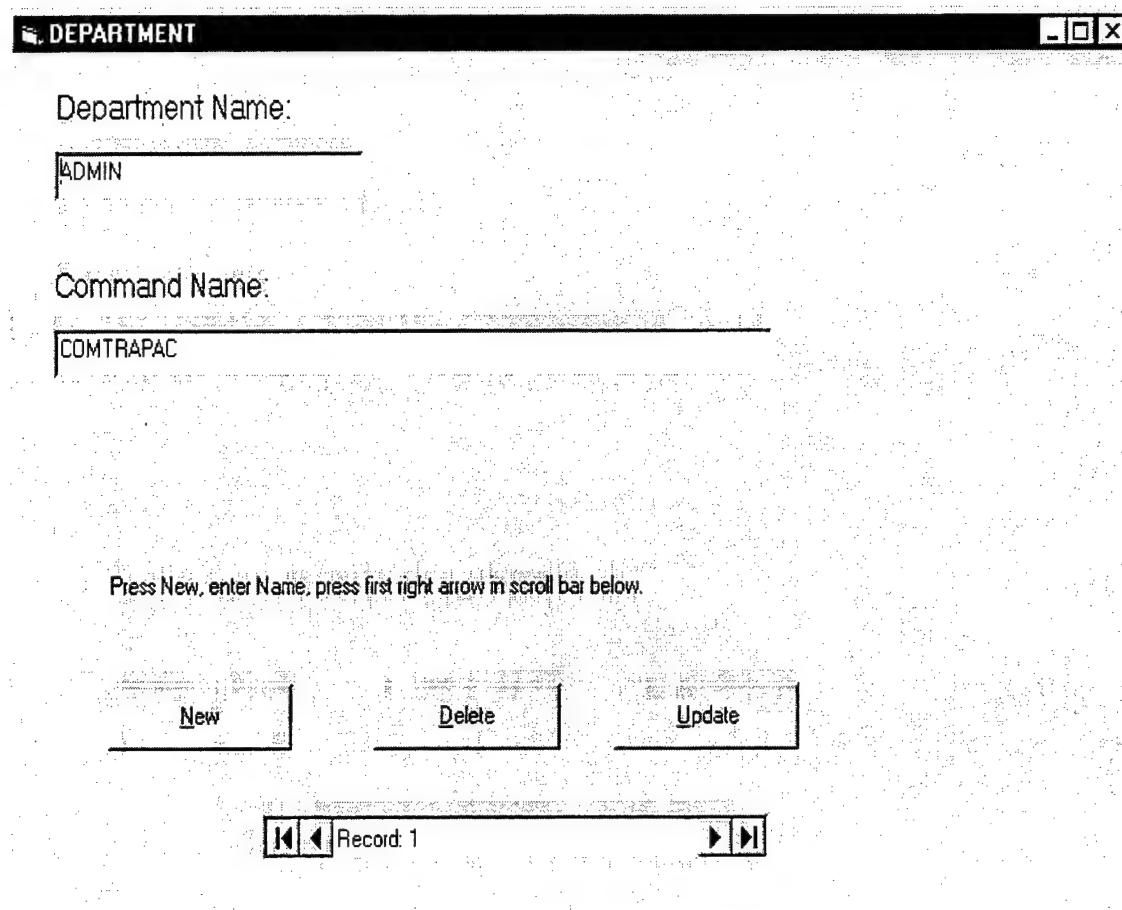
Figure 20. Department Screen

The Main Menu screen allows the user to select among the functions provided by the program, Figure 21. By selecting any of the buttons the user can access the other screens. The PRT Database option allows the user to access the Administer PRT screen. Enter Servicemember allows the user to enter information on each servicemember in his command. The Enter UIC, Department and Update Command allows the user to access the screens described above. The User's Manual button allows the user to view the User Manual for this program. The Command Summary button displays a created Crystal Reports' Command Summary. The Forms button allows the user to select any forms available through this program. The Filters option allows the user to filter records based

41

on PRT number or department. The Merge button allows the user to import old PRT databases. The Exit button ends the program.
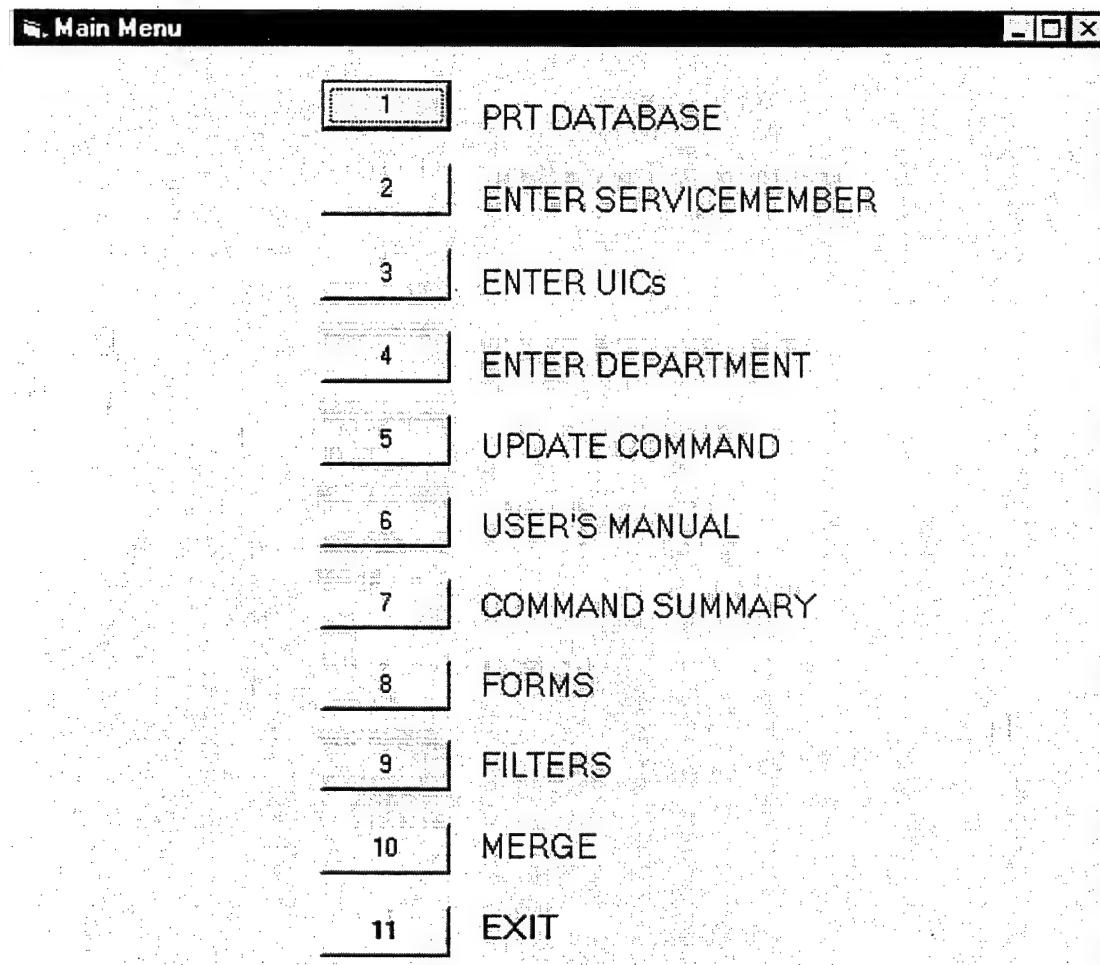
Figure 21. Main Menu Screen

The Administer PRT screen is shown in Figure 22. If the user has previously entered a PRT in the database, that PRT will be listed in the combo box. By selecting the PRT in the combo box and pressing the Go button, the user will be presented the Results Screen with records filtered for the selected PRT. If the user has not entered a PRT previously or desires to enter a new PRT, the New button will display the Blank PRT screen.

42
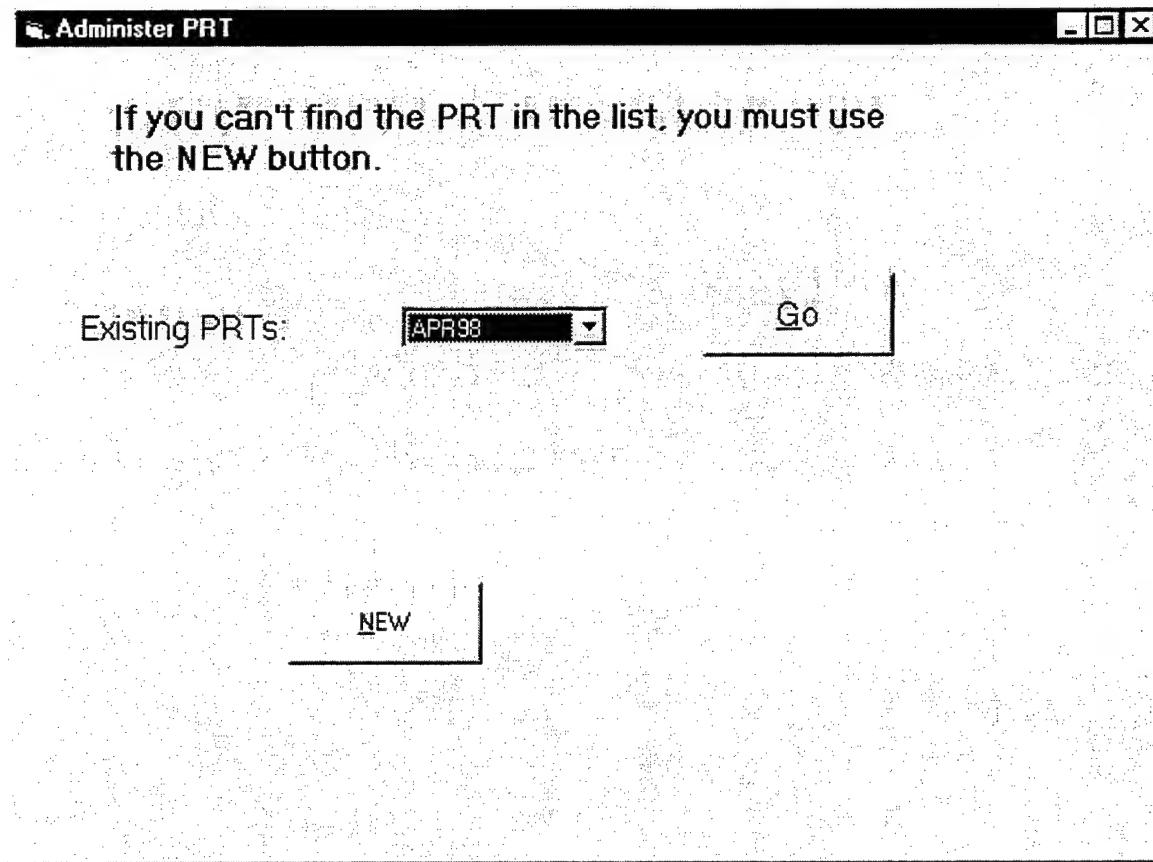
Figure 22. Administer PRT Screen

The New PRT screen is shown in Figure 23. If the user wants to add a new record, he presses the Add button and enters the PRT number and PRT date. By adding a new PRT, new records in the Results table are inserted for all servicemembers who do not have a checked out date in the Servicemember table. The new records consist of the SSN, PRT number, PRT date, and command. The Cancel button hides this form.
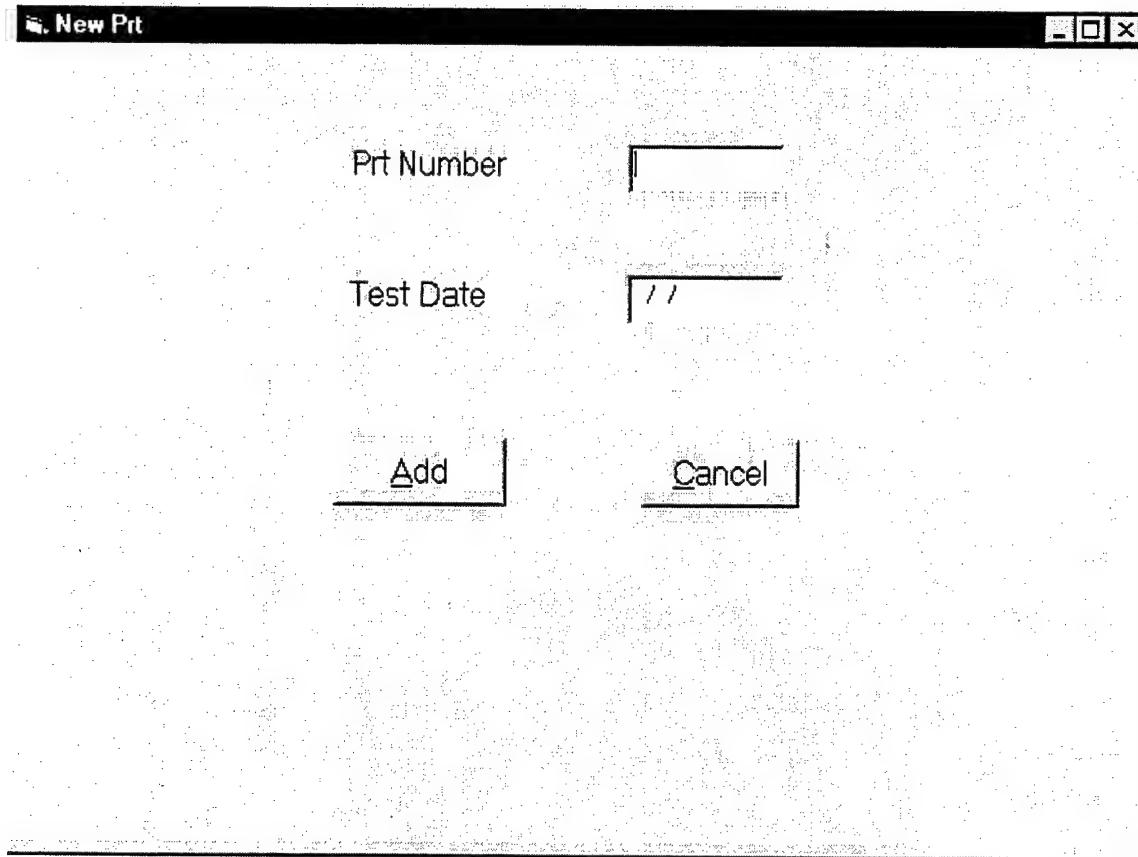
Figure 23. New PRT Screen

The User's Manual screen is depicted in Figure. The File Menu consists of four options open, print, close and exit. The open option will display the Windows file selection screen with text files filtered. The user could then open the file USERSMANUAL.txt, which will be displayed in the text box. Once opened the print and close commands will be enabled. If the user desires to print the file, he can select the Print option. The Close option closes the file and the Exit menu option ends the program.
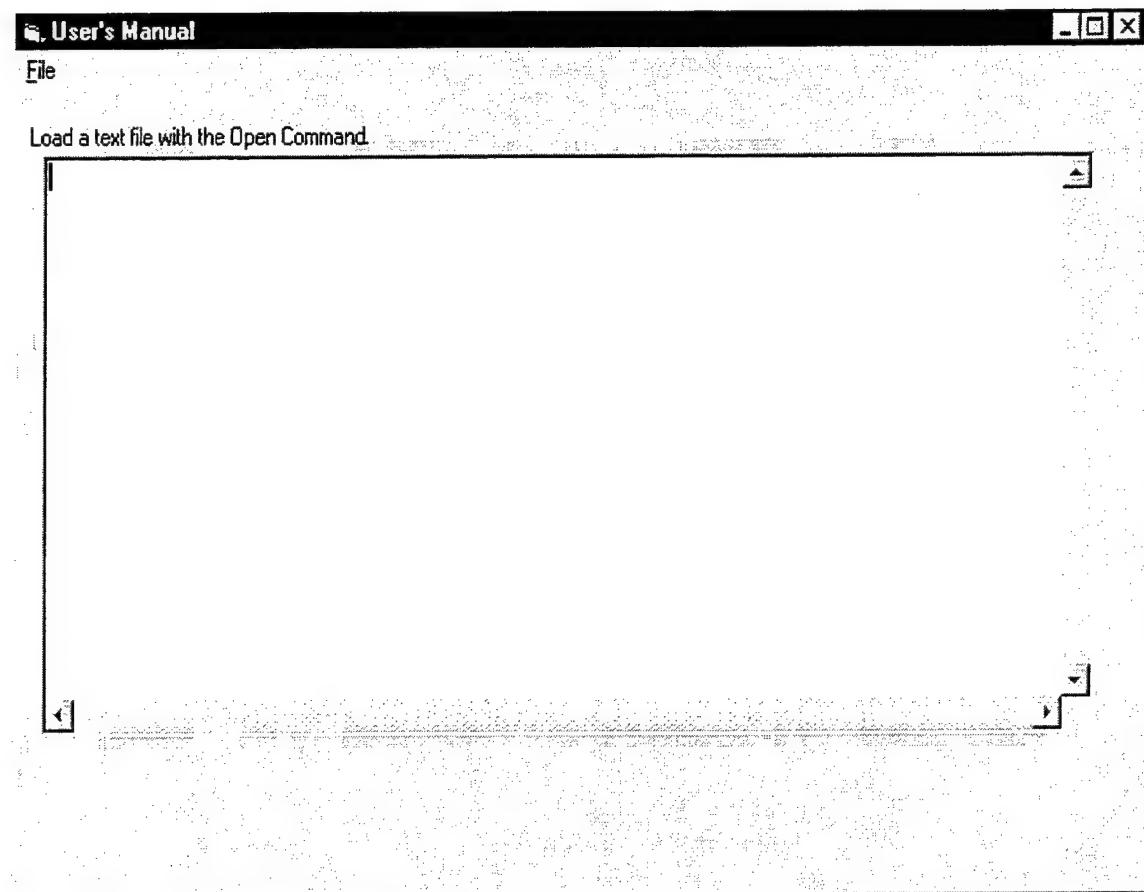
Figure 24. User's Manual Screen

If the user selects the Command Summary button, he will be asked to enter some information such as fiscal year, previous PRT, last PRT and all the information for numbers 8 and 9 of the Command Summary. Once this information is added, the completed Command Summary will be displayed and the screen will allow the user to print it. The preview screen was automatically generated by Crystal Reports 4.6.

The Forms screen is shown in Figure 25. Selecting the LON option, automatically generates a Letter of Notification form for every officer who has failed the

45

latest PRT. The user will preview the report in a form automatically generated by Crystal Reports that allows the user to scroll through the letters and print them.

The Page 13 One/Two Failures and the Page 13 Three Failures options allow the user to print Page 13 for a specific member for the latest PRT. The user will be prompted to enter the SSN. The information regarding body fat and scores will be automatically retrieved from the database. By selecting the Scoring Sheet option, the user will have a form listing all members of his command by SSN and name and columns to enter the scores for each event. As data is entered in the Results table, the information will be updated in this form. Therefore, this form could be used to display results once the PRT is completed. The Advancement Report and the Recommendations based on 4-year History options are listings of all members in the command and displays whether they are recommended for advancement, frocking, etc. The Notice of CDCP is a listing of all members who have either failed the latest PRT, body fat or Sit-And-Reach and should be placed in the Command Directed Conditioning Program (CDCP). The Medical Review Board option allows the user to print a letter recommending a particular individual for a medical review board. Finally, the Award Certificate option automatically generates award certificates for each member who has received an outstanding in the latest PRT. By closing the preview screens, the user will return to the Forms screen. The Exit button in the Forms screen ends the program.
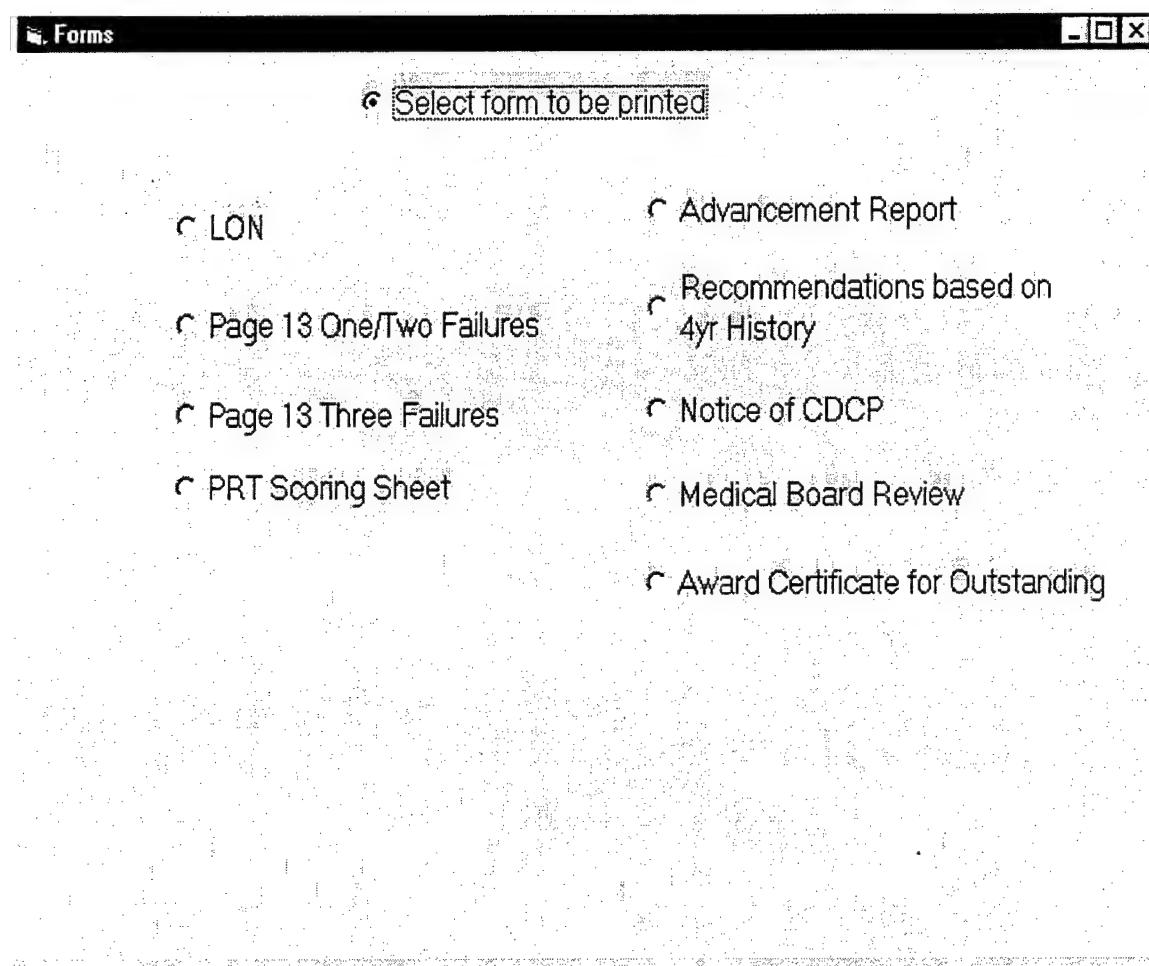
Figure 25. Forms Screen

The Filters screen is shown in Figure 26. When the user selects this option an input box will appear asking for a PRT number, based on this PRT number the records will be filtered. The options available from the combo box are Enlisted, Officer, Failed By Department and Failures. The data grid changes the records according to the selection. No changes can be made in the data grid since it is Read Only. All fields in the connecting table or query will be displayed.
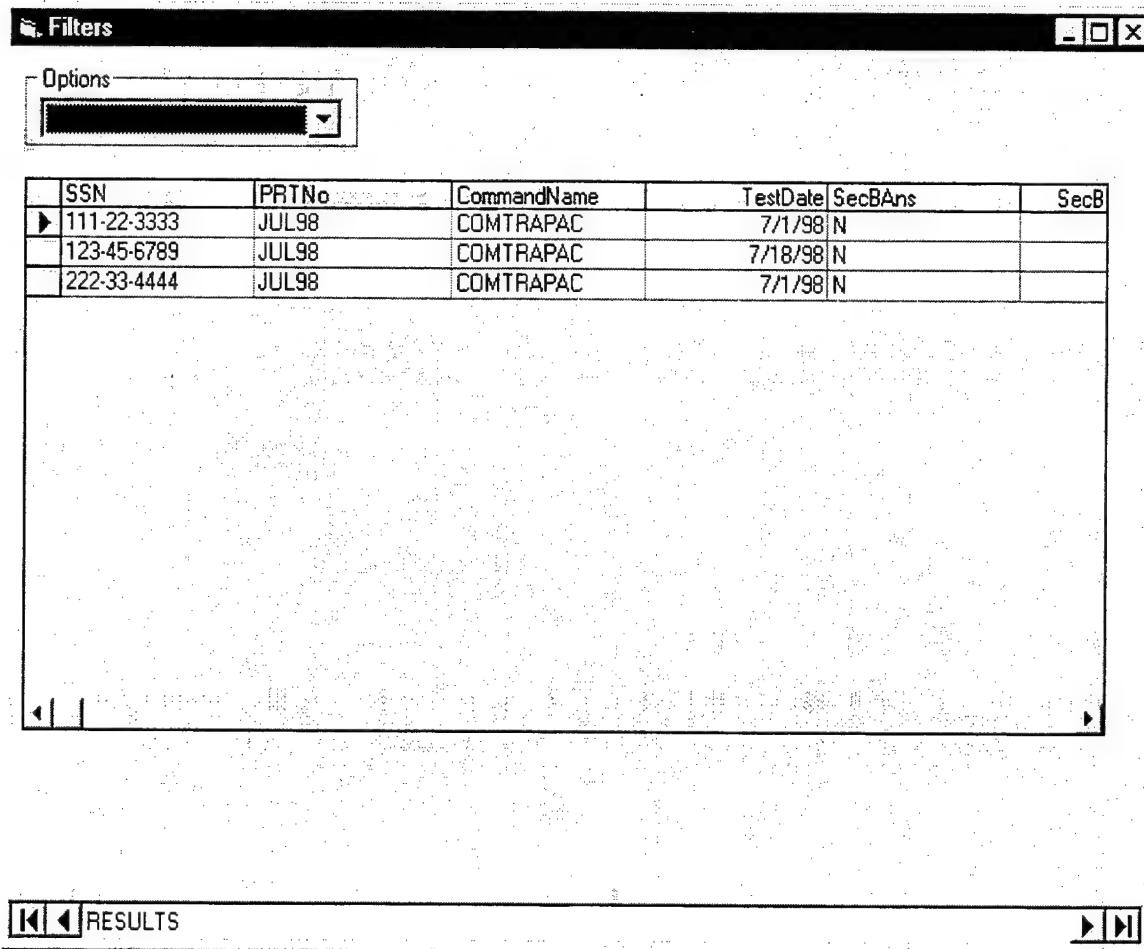
Figure 26. Filters Screen

The Servicemember screen is shown in Figure 27. A navigation control allows the user to scroll through all the records in the Servicemember table. The command name box is automatically filled in based on the command table. The Physical Expires and Aftercare Completed labels are automatically generated by the program. The New button allows the user to add a new record to the database. The Delete button allows the user to delete a record from the database. The Update button allows the user to make

changes to an existing record. The Print Screen button allows the user to print the current record in the screen.

SERVICEMEMBER

SSN:	111-00-7777	Telephone	9-8790
First Name:	c	Checked Aboard:	5/7/98
Middle Initial:	a	Checked Out:	
Last Name:	t	Current Physical:	
Paygrade:	U1	Physical Expires:	
Active Duty Status:	ACT	WMP Sched:	5/6/98
Sex:	M	WMP Completed:	
DOB:	1/1/78	Aftercare Completed:	
Dept Name:	ADMIN		
CommandName:	COMTRAPAC		

New   Delete   Update   PrintScreen

Record: 1

Figure 27. Servicemember Screen

The Results screen is shown in Figure 28. The last name combo box is filled in automatically based on the Servicemember table. When the user adds a new PRT, a new record in the Results table is added and the SSN, PRTNo, test date, and command name are filled in. Therefore the Results screen will display data in these controls. The body fat will be filled in if the measurement fields contain data. The standards label has a default of "WS" if the measurements entered determine the member to be not within standards, a "NS" will be displayed. The points and categories for each event and overall will be displayed when the user adds the scores for each event. The Print Screen button

allows the user to print the current record in the screen. The Delete button allows the user to delete a record in the Results table and the Update button allows a user to modify a record.

Figure 28. Results Screen

The remaining forms were generated by Crystal Reports 6.0 and consist of a preview control that is used to display each report designed with this version of the program.

IV. CONCLUSIONS AND RECOMMENDATIONS

Overall, this research was successful in the development of an improved graphical user interface for the Navy's Physical Readiness Test command database. While perfect and all-inclusive functionality was not achieved, enough functionality was supported to demonstrate the concepts of the improved interface.

Since the aspects of the prototype derived from this research primarily relate to the users of this program, any validation of the improvements made would have to be accomplished with a usability survey of users. Such a survey was not conducted as part of this research, so no conclusions can be stated regarding the effectiveness of the improvements. However, a few personal observations from experience obtained as a Command Fitness Coordinator can be made along with some recommendations for future research.

A. RESULTS OF RESEARCH

Based on personal experience with the 1988 DOS version and the Alpha 4 version, I would submit that the developed prototype is a major improvement, which supports both efficiency and effectiveness.

The prototype provides an easy to use interface to the underlying database. There is no need for the user to learn a Database Management System (DBMS) application. The program's direct interaction with the underlying database is hidden from the user. The user does not have to learn to design a database, use SQL or be concerned with table

51

or query design and organization. By merely using the mouse to press buttons or entering text, the user can utilize all of the available functionality.

This prototype removes many of the limitations of the DOS versions. The records in the database are maintained by SSN. Since the database enforces key constraints, duplicate SSN entries are not allowed unless the user is attempting to add new results for a different PRT. In addition, the database prevents a user from saving a record without the SSN. Like the DOS version, it is still possible to leave the scores fields blank, but once the information is added the points and categories will be generated. Unlike the DOS version, there is no conclusion as to participation in the PRT based on the body fat field. It is possible for a member not to be within standards and still receive a score when tested if cleared by medical.

The program not only provides a technological advancement since 1988 but it also meets current policy and guidance for administration of the PRT. The obese and overfat categories are no longer used. The standards published in OPNAVINST 6110.1e were incorporated into the database. In addition, it is possible for a member to fail the Sit-and-Reach but will not constitute an overall failure. Furthermore, body fat computations are based on the latest formulas.

The prototype also solves some of the limitations of the Alpha 4 program. The user does not have to be familiar with SQL. The menus, options and fields are self-explanatory and what might be unclear can be resolved by accessing the detailed user manual included as an option in the Main Menu screen. In addition, the user can save up to 1 gigabyte of data in the underlying prt.mdb file. The prototype does away with saving

individual databases. It contains all of its records in one centralized database. Moreover, there is no need to set up printers, add link fields, or use the arrow keys.

The program had to be easily distributed and installed at a low cost. The executable program and required files were included in a setup program which can be distributed on 3 ½" disks. The user will be able to run the setup program as he would any other Windows setup program.

The greatest improvement of the prototype was the ability to achieve the last requirements of the research goal: provide a wide range of tailored statistical data and reports, reduce the administrative burden on the CFC, provide a rapid individual or command status, and import existing PRT data. The program automatically computes all the statistics necessary to complete and generate the Command Summary. In addition, there are 9 reports available from the Forms screen based on the latest PRT. All Letters of Notification and Award Certificates are generated for every member who failed the latest PRT or received Outstanding respectively. Pages 13 for one, two or three failures can be generated by entering the member's SSN. The PRT Scoring Sheet can be used as a tool to record scores or to view results. The Advancement Report, Recommendations based on 4-year History and Notice of CDCP provide a listing of recommendations for administrative actions and listing of everyone who should attend the Command Directed Conditioning Program. The last form, the Medical Review Board letter can be generated for any member by adding name and medical institution information.

A rapid individual or command status can be viewed by using the Filters screen. Through this screen, a user can view all the records, officers' records, enlisted records,

failures by department or all failures by entering the selected PRT or department as prompted by the program.

Finally, the program can easily merge previous records from the 1988 version by simply pressing a button. The data will automatically be added to the centralized database. The user will only have to update the servicemember and results fields as necessary.

As previously mentioned, not all desired features were implemented by the graphical user interface. The basic limitation was time constraints. Error checking for input data is very limited. In addition, the program does not compute the number of consecutive medical waivers nor does it determine if the member is cleared to take the PRT. These decisions will have to be made by the CFC based on the records. Furthermore, the list of generated reports is not all-inclusive. It does generate the forms required by OPNAVINST 6110.1e but it is limited in the number of other reports that may assist the CFC in doing his job. These limitations are not critical to the use of the program. Provided the user enters the data as outlined in the user's manual, the likelihood of the program failing is minimal. The result is that the prototype provides sufficient capabilities to be utilized until a more robust implementation is available. In addition, since the application and database are separate, it is possible to build an entirely new graphical user interface without needing to redesigned the underlying database.

B. CRITIC OF RESEARCH

As with any program, criticism can be made of the design and implementation. This program offers no exception. All of the problems, both in design and implementation were mine to address. Time constraints and unfamiliarity with the

54

programs used to develop the prototype prevented me from developing a perfect and robust interface.

1. User Interface

The timeliness of the screens displayed as the user presses the buttons is directly affected by the operations that are performed prior to displaying the forms. Those forms that must be filtered or must perform a query take longer to be displayed. This is the case of the Filters screen and the Results screen. By contrast, those forms that display tables in the database such as the Servicemember screen are displayed very quickly.

Reports may be viewed even though they contain no data. In some instances although the report contains only one record, two forms are generated. The user must use the scroll arrow in the form to view the desired form letter. However, this limitation does not prevent the user from printing the forms. I attribute this problem to my lack of knowledge on working with Crystal Reports.

Microsoft Windows design conventions were not used. The buttons are placed as I saw fit. The screens are not of the same size. Although I tried to maintain some consistency, there were no conventions in the size of fonts or objects within the screens.

Most of the windows are displayed in a modeless form that means the user can have several windows opened at the same time. The only exceptions are the Command, UIC, Department screens upon initialization of the database and the NewPRT screen. By default all message and input boxes are modal.

2. Implementation

The prototype was implemented using Visual Basic 5.0 with Service Pack 3, Crystal Reports 4.6 and 6.0, and Access 97. The first step in developing the interface was to design the database. This was simplified by having a thorough conceptual design. The translation between the ER diagram and the tables in the database was not as direct as Reference 9 specified. I found that queries that contained several inner joins ran extremely slowly. To improve performance and whenever possible, queries were designed with tables and a limited number of queries. At times, additional fields were added to the queries so that they could be used in nested queries even though the fields did not directly relate to the information being withdrawn from the query. Building the queries and tables was not a difficult task due to the ability to switch between design and results. The most difficult problem encountered was the implementation of the Run/Swim Time. The type provided by Access is a date/time type. Time consists of hour, minutes, seconds and AM or PM. Since I only needed to record the minutes and seconds, I restricted the displayed format in both the field and the table containing the time values and categories. However, when I ran the code in Visual Basic, time displayed as 12:00:00AM or a date. To correct this problem, the formatting was done in the code that displays or receives the time value.

The coding performed in Visual Basic was minimal since the program automatically generates most of the underlying code and the Microsoft Jet engine links to the database. The use Wizards to generate a database application was a good initial frame of reference. Once a Wizard created some of the forms, it was easier to design the remaining forms. Although the Jet engine links the program to the database, I found that

performance improved once I had an ODBC connection set up in the control panel. However, several times the Visual Basic application crashed for no apparent reason. The error given was an "out of stack" problem or "not enough virtual memory" when the computer had been set to its maximum of 93 paging files. To limit the amount of coding, I took advantage of using the design time properties and the data controls and avoiding the use of DAO unless necessary. An experienced programmer will probably opt for more DAO usage.

The use of Crystal Reports is encouraged by Visual Basic to build reports. Crystal Reports 4.6 is available with Visual Basic 5.0 and can be used to design primarily summary or listings reports. A report designed with Crystal Reports 4.6 must be accessed by code. It does not generate a viewing form as does Crystal Reports 6.0. However, I found that Crystal Reports 4.6 was easier to use since it provided a preview screen which Crystal Reports 6.0 lacked. In addition, adding and deleting queries or tables from a report was easier with 4.6. The primary advantage of Crystal Reports 6.0 was the ability to generate a form letter which was used for the Letter of Notification form and it implements a text object with properties which makes it easy to insert a text string using an input box. In the 4.6 version, the designer must declare a formula box to insert a text string and format the string using code. Most of the forms were generated with Crystal Reports 6.0. I experienced a limit in the number of reports I could design with Crystal Reports 6.0. After the sixth designed report, it was my intention to design a form letter for personnel assigned to the Command Directed Conditioning Program but every time I tried to add the report, the Visual Basic application crashed. A Crystal

Reports engineer at Seagate claimed that there were no limits in the number of forms and that it must have been a system limitation. I was never able to solve this problem.

C. RECOMMENDATIONS FOR FUTURE RESEARCH

In order to design a more functional and robust program than the one accomplished by this research, several ideas came to mind, most of which occured much too late to be incorporated into this prototype. First and foremost, there should be error checking on all information entered by the user. Second, the screen layout design should follow published Windows design conventions. Third, the program should compute number of consecutive medical waivers and display a warning message and automatically generate the recommendation for medical board letter. In addition, even though this prototype computes the number of consecutive failures in four years, it should also display a warning message and automatically generate the page 13 notifications and any other administrative form, such as notification of administrative board procedures. As Chief Gissel's program, it should determine whether or not the member is cleared for the PRT and a listing for administrative purposes listing pass/fail and standards for members of the command.

Finally, since it is envisioned that this program become a centralized database, it should be converted to an internet/intranet application. Such an application should permit the transmission of individual records using a secure method over the internet or using e-mail. Such a program must also achieve the elimination of the pink folder and the requirement to physically transfer the folder upon detachment from a command.

58

Although primarily designed for use with a mouse, I tried in most cases to add control+key shortcuts. However, it will be a lot easier to use a pointing device with this prototype.

```
Global gsDatabase As String
Global gsConnect As String
Global gsRecordsource As String


Sub Main()
    gsDatabase = App.Path & "\prt.mdb"
    gsConnect = ""
End Sub
```

'Welcome Screen
```
Private Sub Command1_Click()
    'Delete Results older than 4 years
    Dim db As Database


    'Check to see if command is blank
        Load Command
        Load UIC
        Load DEPARTMENT
    If  (Command![Text1].Text  =  "")  Or  (UIC![Text1].Text  =  "")  Or
(DEPARTMENT![Text1].Text = "") Then
        MsgBox ("Please enter Command, UIC, and Department information."), ,
"Initialize Database"
        Welcome.Hide
        Command.Command2 = True
        Command.Label8 = "Press Add button and enter information."
        Command.Show 1       'modal form
        UIC.Show 1           'modal form
        DEPARTMENT.Show 1    'modal form


    Else
```

61

```
        With Me.dtaPRT
          Set db = .Database
          db.Execute "DeleteOldResults"
        End With
        Welcome.Hide


      End If


        MainMenu.Show
    End Sub


    Private Sub Command2_Click()
      End
    End Sub


    Private Sub Form_Load()
      dtaPRT.DatabaseName = App.Path & "\prt.mdb"
    End Sub
```

'User Manual Screeen

```
Private Sub Command1_Click()
    'Delete Results older than 4 years
    Dim db As Database



    'Check to see if command is blank
      Load Command
      Load UIC
      Load DEPARTMENT
    If   (Command![Text1].Text   =   "")   Or   (UIC![Text1].Text   =   "")   Or
(DEPARTMENT![Text1].Text = "") Then
        MsgBox ("Please enter Command, UIC, and Department information."), ,
"Initialize Database"
        Welcome.Hide
        Command.Command2 = True
        Command.Label8 = "Press Add button and enter information."
```

```vb
            Command.Show 1        'modal form
            UIC.Show 1            'modal form
            DEPARTMENT.Show 1     'modal form

      Else
        With Me.dtaPRT
          Set db = .Database
          db.Execute "DeleteOldResults"
        End With
        Welcome.Hide


      End If
        MainMenu.Show
    End Sub


    Private Sub Command2_Click()
      End
    End Sub


    Private Sub Form_Load()
      dtaPRT.DatabaseName = App.Path & "\prt.mdb"
    End Sub
```

'UIC Screen
```vb
Dim commandName As String


Private Sub Command4_Click()
  End
End Sub


Private Sub Form_Activate()
  Data2.Recordset.MoveFirst
  commandName = Data2.Recordset!commandName
End Sub


Private Sub Command1_Click()
```

```
      Data1.Recordset.AddNew
      Text2.Text = commandName
   End Sub


Private Sub Command2_Click()
    'this may produce an error if you delete the last
    'record or the only record in the recordset
    With Data1.Recordset
       .Delete
       .MoveNext
       If .EOF Then .MoveLast
    End With
End Sub


Private Sub Command3_Click()
   Data1.UpdateRecord
   Data1.Recordset.Bookmark = Data1.Recordset.LastModified
End Sub


Private Sub Data1_Error(DataErr As Integer, Response As Integer)
    'This is where you would put error handling code
    'If you want to ignore errors, comment out the next line
    'If you want to trap them, add code here to handle them
    MsgBox "Data error event hit err:" & Error$(DataErr), , "UIC Error"
    Response = 0   'throw away the error
End Sub


Private Sub Data1_Reposition()
    Screen.MousePointer = vbDefault
    On Error Resume Next
    'This will display the current record position
    'for dynasets and sanpshots
    Data1.Caption = "Record: " & (Data1.Recordset.AbsolutePosition + 1)
    'for the table object you must set the index property when
    'the recordset gets created and use the following line
```

```vb
        Data1.Caption  =   "Record:   "   &   (Data1.Recordset.RecordCount   *
(Data1.Recordset.PercentPosition * 0.01)) + 1
        End Sub


        Private Sub Data1_Validate(Action As Integer, Save As Integer)
            'This is where you put validation code
            'This event gets called when the following actions occur
            Select Case Action
                Case vbDataActionMoveFirst
                Case vbDataActionMovePrevious
                Case vbDataActionMoveNext
                Case vbDataActionMoveLast
                Case vbDataActionAddNew
                Case vbDataActionUpdate
                Case vbDataActionDelete
                Case vbDataActionFind
                Case vbDataActionBookmark
                Case vbDataActionClose
                    Screen.MousePointer = vbDefault
            End Select
            Screen.MousePointer = vbHourglass
        End Sub


        Private Sub Form_Load()
            Data1.DatabaseName = App.Path & "\prt.mdb"
            Data2.DatabaseName = App.Path & "\prt.mdb"
        End Sub


        Private Sub Form_Unload(Cancel As Integer)
            Screen.MousePointer = vbDefault
        End Sub


--------------------------------------------------------------------------------------
' Servicemember Screen
Dim commandName As String
```

65

```vb
Private Sub Command4_Click()
    End
End Sub


Private Sub Form_Activate()
    Data2.Recordset.MoveFirst
    commandName = Data2.Recordset!commandName
End Sub


Private Sub Command1_Click()
    Data1.Recordset.AddNew
    Text2.Text = commandName
End Sub


Private Sub Command2_Click()
    'this may produce an error if you delete the last
    'record or the only record in the recordset
    With Data1.Recordset
        .Delete
        .MoveNext
        If .EOF Then .MoveLast
    End With
End Sub


Private Sub Command3_Click()
    Data1.UpdateRecord
    Data1.Recordset.Bookmark = Data1.Recordset.LastModified
End Sub


Private Sub Data1_Error(DataErr As Integer, Response As Integer)
    'This is where you would put error handling code
    'If you want to ignore errors, comment out the next line
    'If you want to trap them, add code here to handle them
    MsgBox "Data error event hit err:" & Error$(DataErr), , "UIC Error"
    Response = 0  'throw away the error
End Sub
```

```vb
Private Sub Data1_Reposition()
    Screen.MousePointer = vbDefault
    On Error Resume Next
    'This will display the current record position
    'for dynasets and sanpshots
    Data1.Caption = "Record: " & (Data1.Recordset.AbsolutePosition + 1)
    'for the table object you must set the index property when
    'the recordset gets created and use the following line
    'Data1.Caption = "Record: " & (Data1.Recordset.RecordCount *
(Data1.Recordset.PercentPosition * 0.01)) + 1
End Sub


Private Sub Data1_Validate(Action As Integer, Save As Integer)
    'This is where you put validation code
    'This event gets called when the following actions occur
    Select Case Action
        Case vbDataActionMoveFirst
        Case vbDataActionMovePrevious
        Case vbDataActionMoveNext
        Case vbDataActionMoveLast
        Case vbDataActionAddNew
        Case vbDataActionUpdate
        Case vbDataActionDelete
        Case vbDataActionFind
        Case vbDataActionBookmark
        Case vbDataActionClose
            Screen.MousePointer = vbDefault
    End Select
    Screen.MousePointer = vbHourglass
End Sub

Private Sub Form_Load()
    Data1.DatabaseName = App.Path & "\prt.mdb"
    Data2.DatabaseName = App.Path & "\prt.mdb"
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
```

'Administer PRT form
```
Private Sub Command1_Click()
    frmBlankPrt.Show 1
End Sub


Private Sub Command2_Click()
    Load frmRESULTS

    frmRESULTS.Caption = DBCombo1.Text & " Results"
    frmRESULTS.Data1.Recordset.Filter = "PRTNo = '" & DBCombo1.Text & "'"
    frmRESULTS.Data1.Recordset.OpenRecordset

    Dim rs As Recordset
    Set rs = frmRESULTS.Data1.Recordset.OpenRecordset
    Set frmRESULTS.Data1.Recordset = rs

    frmRESULTS.Data6.Recordset.Filter = "PRTNo = '" & DBCombo1.Text & "'"
    frmRESULTS.Data6.Recordset.OpenRecordset
    Set rs = frmRESULTS.Data6.Recordset.OpenRecordset
    Set frmRESULTS.Data6.Recordset = rs

    frmRESULTS.Show
    selPrtfrm.Hide
End Sub


Private Sub Command3_Click()
    End
End Sub
```

68

```vb
Private Sub Form_Activate()
    Data1.Refresh
End Sub


Private Sub Form_Load()
    Data1.DatabaseName = App.Path & "\prt.mdb"
    Data2.DatabaseName = App.Path & "\prt.mdb"
End Sub
```

'**Main Menu Screen**

```vb
Option Explicit


Private Sub Command1_Click()
    selPrtfrm.Show
End Sub


Private Sub Command10_Click()
    DEPARTMENT.Show
End Sub


Private Sub Command11_Click()
    Command.Show
End Sub


Private Sub Command2_Click()
    UserMan.Show
End Sub


Private Sub Command3_Click()
    Dim db As Database
    Dim rs As Recordset
    Dim nCount As Long
    Dim inpString As String
    Dim inpDate As String

    With Me.crptPRT
```

```vb
Dim i As Integer
For i = 0 To 37
   .Formulas(i) = ""
Next i



.ReportFileName = App.Path & "\summary.rpt"
.WindowTitle   =   "COMMAND   PHYSICAL   READINESS   TEST
SUMMARY"
.WindowState = crptMaximized
.Connect = "DSN=PRT"


Set db = Me.Data1.Database


'Officer1
Set rs = db.OpenRecordset("TotalOfficer")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)        '0 - first, 1 - 2 second field, 2 - third field, ...
End If
rs.Close


.Formulas(0) = "officer1 = " & nCount      'Officer1 is the formula name in
the crystal report


'Enlisted1
Set rs = db.OpenRecordset("TotalEnlisted")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)
End If
rs.Close
```

70

```vb
        .Formulas(1) = "Enlisted1 = " & nCount        ' Enlisted1 is the formula name
in the crystal report

        'Officer2
        Set rs = db.OpenRecordset("TotalOTakingPRT")
        If rs.EOF Then
            nCount = 0
        Else
            nCount = rs(0)        ' 0 - first, 1 - 2 second field, 2 - third field, ...
        End If
        rs.Close


        .Formulas(2) = "Officer2 = " & nCount
        'Enlisted2
        Set rs = db.OpenRecordset("TotalETakingPRT")
        If rs.EOF Then
            nCount = 0
        Else
            nCount = rs(0)
        End If
        rs.Close


        .Formulas(3) = "Enlisted2 = " & nCount
        'Officer3
        Set rs = db.OpenRecordset("TotalMedicalWaiverOfficer")
        If rs.EOF Then
            nCount = 0
        Else
            nCount = rs(0)        ' 0 - first, 1 - 2 second field, 2 - third field, ...
        End If
        rs.Close


        .Formulas(4) = "Officer3 = " & nCount
        'Enlisted3
        Set rs = db.OpenRecordset("TotalMedicalWaiverEnlisted")
        If rs.EOF Then
```

71

```vb
   nCount = 0
Else
   nCount = rs(0)
End If
rs.Close


.Formulas(5) = "Enlisted3 = " & nCount
'Officer4
Set rs = db.OpenRecordset("TotalONotTestedOver50")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)        '0 - first, 1 - 2 second field, 2 - third field, ...
End If
rs.Close


.Formulas(6) = "Officer4 = " & nCount
'Enlisted4
Set rs = db.OpenRecordset("TotalENotTestedOver50")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)
End If
rs.Close


.Formulas(7) = "Enlisted4 = " & nCount
'Officer5
Set rs = db.OpenRecordset("TotalONotTestedOther")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)        '0 - first, 1 - 2 second field, 2 - third field, ...
End If
rs.Close
```

```
.Formulas(8) = "Officer5 = " & nCount
'Enlisted5
Set rs = db.OpenRecordset("TotalENotTestedOther")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)
End If
rs.Close


.Formulas(9) = "Enlisted5 = " & nCount
'Officer6
Set rs = db.OpenRecordset("TotalOPassingPRT")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)          '0 - first, 1 - 2 second field, 2 - third field, ...
End If
rs.Close


.Formulas(10) = "Officer6 = " & nCount
'Enlisted6
Set rs = db.OpenRecordset("TotalEPassingPRT")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)
End If
rs.Close


.Formulas(11) = "Enlisted6 = " & nCount
'Officer7
Set rs = db.OpenRecordset("TotalOFailingSitReach")
If rs.EOF Then
   nCount = 0
Else
```

73

```
        nCount = rs(0)          '0 - first, 1 - 2 second field, 2 - third field, ...
End If
rs.Close


.Formulas(12) = "Officer7 = " & nCount
'Enlisted7
Set rs = db.OpenRecordset("TotalEFailingSitReach")
If rs.EOF Then
    nCount = 0
Else
    nCount = rs(0)
End If
rs.Close


.Formulas(13) = "Enlisted7 = " & nCount
'Officer8
Set rs = db.OpenRecordset("TotalOFailingCurlUps")
If rs.EOF Then
    nCount = 0
Else
    nCount = rs(0)          '0 - first, 1 - 2 second field, 2 - third field, ...
End If
rs.Close


.Formulas(14) = "Officer8 = " & nCount
'Enlisted8
Set rs = db.OpenRecordset("TotalEFailingCurlUps")
If rs.EOF Then
    nCount = 0
Else
    nCount = rs(0)
End If
rs.Close


.Formulas(15) = "Enlisted8 = " & nCount
'Officer9
```

```vb
Set rs = db.OpenRecordset("TotalOFailingPushUps")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)          '0 - first, 1 - 2 second field, 2 - third field, ...
End If
rs.Close


.Formulas(16) = "Officer9 = " & nCount
'Enlisted9
Set rs = db.OpenRecordset("TotalEFailingPushUps")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)
End If
rs.Close


.Formulas(17) = "Enlisted9= " & nCount
'Officer10
Set rs = db.OpenRecordset("TotalOFailingRunSwim")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)          '0 - first, 1 - 2 second field, 2 - third field, ...
End If
rs.Close


.Formulas(18) = "Officer10 = " & nCount
'Enlisted10
Set rs = db.OpenRecordset("TotalEFailingRunSwim")
If rs.EOF Then
   nCount = 0
Else
   nCount = rs(0)
End If
```

```
        rs.Close

        .Formulas(19) = "Enlisted10 = " & nCount
        'Officer11
        Set rs = db.OpenRecordset("TotalOFailedBodyFat")
        If rs.EOF Then
            nCount = 0
        Else
            nCount = rs(0)        '0 - first, 1 - 2 second field, 2 - third field, ...
        End If
        rs.Close


        .Formulas(20) = "Officer11 = " & nCount
        'Enlisted11
        Set rs = db.OpenRecordset("TotalEFailedBodyFat")
        If rs.EOF Then
            nCount = 0
        Else
            nCount = rs(0)
        End If
        rs.Close


        .Formulas(21) = "Enlisted11 = " & nCount
        'Officer12
        nCount = Val(InputBox("Enter number of officers not recommended for
frocking for failure of PRT."))
        .Formulas(22) = "Officer12 = " & nCount
        nCount = 0
        'Enlisted12
        nCount = Val(InputBox("Enter number of enlisted not recommended for
frocking for failure of PRT."))
        .Formulas(23) = "Enlisted12 = " & nCount
        nCount = 0
        'Officer13
        nCount = Val(InputBox("Enter number of officers not recommended for
promotion/advancement for failure of PRT."))
```

```
        .Formulas(24) = "Officer13 = " & nCount
        nCount = 0
        'Enlisted13
        nCount = Val(InputBox("Enter number of enlisted not recommended for
promotion/advancement for failure of PRT."))
        .Formulas(25) = "Enlisted13 = " & nCount
        nCount = 0
        'Officer14
        nCount = Val(InputBox("Enter number of officers not recommended for
reenlistment/redesignation for failure of PRT."))
        .Formulas(26) = "Officer14 = " & nCount
        nCount = 0
        'Enlisted14
        nCount = Val(InputBox("Enter number of enlisted not recommended for
reenlistment/redesignation for failure of PRT."))
        .Formulas(27) = "Enlisted14 = " & nCount
        nCount = 0
        'Officer15
        nCount = Val(InputBox("Enter number of officers not recommended for
frocking for failure of Body Fat."))
        .Formulas(28) = "Officer15 = " & nCount
        nCount = 0
        'Enlisted15
        nCount = Val(InputBox("Enter number of enlisted not recommended for
frocking for failure of Body Fat."))
        .Formulas(29) = "Enlisted15 = " & nCount
        nCount = 0
        'Officer16
        nCount = Val(InputBox("Enter number of officers not recommended for
promotion/advancement for failure of Body Fat."))
        .Formulas(30) = "Officer16 = " & nCount
        nCount = 0
        'Enlisted16
        nCount = Val(InputBox("Enter number of enlisted not recommended for
promotion/advancement for failure of Body Fat."))
        .Formulas(31) = "Enlisted16 = " & nCount
```

```
            nCount = 0
            'Officer17
            nCount = Val(InputBox("Enter number of officers not recommended for
reenlistment/redesignation for failure of Body Fat."))
            .Formulas(32) = "Officer17 = " & nCount
            nCount = 0
            'Enlisted17
            nCount = Val(InputBox("Enter number of enlisted not recommended for
reenlistment/redesignation for failure of Body Fat."))
            .Formulas(33) = "Enlisted17 = " & nCount
            nCount = 0

            'set FY
            inpString = InputBox("Enter Fiscal Year. Ex. FY98")
            .Formulas(34) = "FY = " & Chr(34) & inpString & Chr(34)
            inpString = ""
            'set Date of Official PRT
            inpDate = InputBox("Enter Date of Official PRT. mm/dd/yy")
            .Formulas(35) = "DatePRT = " & Chr(34) & inpDate & Chr(34)
            inpDate = ""
            'set Date of Previous official PRT
            inpDate = InputBox("Enter Date of Previous Official PRT. mm/dd/yy")
            .Formulas(36) = "DatePreviousPRT = " & Chr(34) & inpDate & Chr(34)
            inpDate = ""
            .Action = 1
        End With

    End Sub

    Private Sub Command4_Click()
        frmMenu.Show
    End Sub
    Function getField(size As Integer, fileno As Integer) As String
        Dim sLen As Integer
        Dim temp As String
        sLen = Asc(Input(1, fileno))
```

```
        getField = Input$(sLen, fileno)
        temp = Input$(size - sLen, fileno)
End Function



Private Sub Command5_Click()
    Dim dbs As Database
    Set dbs = Workspaces(0).Databases(0)
    Dim Person As Recordset, PRT As Recordset, Results As Recordset
    Dim Command As Recordset, Dept As Recordset, UIC As Recordset

    Set Person = dbs.OpenRecordset("Servicemember", dbOpenDynaset)
    Set PRT = dbs.OpenRecordset("PRT", dbOpenDynaset)
    Set Results = dbs.OpenRecordset("Results", dbOpenDynaset)
    Set Dept = dbs.OpenRecordset("Department", dbOpenDynaset)
    Set Command = dbs.OpenRecordset("Command", dbOpenTable)

    Dim fLen As Integer

'   File dialog box for user to selct file to import
'   Get string back from choose file dialog box to use
'   in the FileLen, and Open statements below
    CommonDialog1.ShowOpen       'display Open dialog box
    On Error GoTo Didnotmerge
    If CommonDialog1.filename <> "" Then
        UserMan.MousePointer = 11   'display hourglass

        fLen = FileLen(CommonDialog1.filename)


        Dim commandName As String  'get command from command table
        Command.MoveFirst
        commandName = Command!commandName


        Open CommonDialog1.filename For Binary Access Read Lock Read As #1
        ' Close before reopening in another mode.
```
79

```
Dim temp As String
Dim fName As String, lName As String, runTime As String
Dim waist As String, sRate As String, hip As String
Dim neck As String, sHeight As String, bodyFat As String
Dim sex As String, sitReach As String, passFail As String
Dim division As String, DEPARTMENT As String, SSN As String
Dim bday As String, prtday As String, situps As String
Dim pushups As String, score As String, weight As String
Dim RunSwim As String
Dim waiverCode As Long
Dim assignInt As Integer
Dim mySitups As Integer
Dim myPushups As Integer


assignInt = 0

Do While fLen >= 181

  lName = getField(15, 1)
  fName = getField(15, 1)
  temp = InputB$(6, 1)
  runTime = getField(5, 1)
  waist = getField(5, 1)
  sRate = getField(5, 1)


  hip = getField(5, 1)
  neck = getField(5, 1)
  sHeight = getField(5, 1)
  bodyFat = getField(2, 1)

  temp = InputB$(8, 1)
  RunSwim = Input$(1, 1)
  'Convert type to Run or SwimMtrs
  If RunSwim = "R" Then
```

```
        RunSwim = "Run"
    Else
        RunSwim = "SwimMtrs"
    End If
    sex = getField(1, 1)
    sitReach = getField(1, 1)
    passFail = getField(1, 1)
    temp = InputB$(2, 1)
    division = getField(15, 1)
    DEPARTMENT = getField(15, 1)
    SSN = getField(11, 1)
    Dim bdate As Date
    bday = getField(8, 1)
    bdate = DateValue(bday)
    prtday = getField(8, 1)
    'get me a PRTNo
    Dim prtdate As Date
    prtdate = DateValue(prtday)
    Dim myPRTNo As String
    myPRTNo = UCase(Format(prtdate, "mmmyy"))
    situps = getField(3, 1)
    pushups = getField(3, 1)
    score = getField(3, 1)
    weight = getField(3, 1)
    temp = InputB$(9, 1)

    'check for w and add waiver codes
    'waived from all events
    If runTime = "W" And sitReach = "W" And situps = "W" And pushups =
"W" Then

        runTime = ""
        sitReach = ""
        mySitups = assignInt
        myPushups = assignInt
        waiverCode = 9
    ElseIf situps = "W" And pushups = "W" Then
```

81

```vb
                                'waived from situps and pushups
                                mySitups = assignInt
                                myPushups = assignInt
                                waiverCode = 6
                        ElseIf (situps = "W" And runTime = "W") Then
                                mySitups = assignInt
                                runTime = ""
                                waiverCode = 7
                        ElseIf pushups = "W" And runTime = "W" Then
                                myPushups = assignInt
                                runTime = ""
                                waiverCode = 8
                        ElseIf runTime = "W" And RunSwim = "Run" Then
                                runTime = ""
                                waiverCode = 4
                        ElseIf runTime = "W" And RunSwim = "SwimMtrs" Then
                                runTime = ""
                                waiverCode = 5
                        ElseIf sitReach = "W" Then
                                sitReach = ""
                                waiverCode = 1
                        ElseIf situps = "W" Then
                                mySitups = assignInt
                                waiverCode = 2
                        ElseIf pushups = "W" Then
                                myPushups = assignInt
                                waiverCode = 3
                        End If 'final if



        '       need to add PRT first for referential integrity
                PRT.FindFirst "PRTNo = '" & myPRTNo & "' AND CommandName = '"
& commandName & "'"
                        If PRT.NoMatch Then
                                With PRT
                                        ' Add new record to end of Recordset object.
```

82

```
            .AddNew
            !TestDate = prtdate
            !commandName = commandName
            !PRTNo = myPRTNo
            .Update              ' Save changes.
          End With
        End If


'       make sure deptname in db
        Dept.FindFirst  "DeptName  =  "'  &  DEPARTMENT  &  "'  AND
CommandName = "' & commandName & ""'"
          If Dept.NoMatch Then
            With Dept
             ' Add new record to end of Recordset object.
             .AddNew
             !DEPARTMENT = DEPARTMENT
             !commandName = commandName
             .Update              ' Save changes.
            End With
          End If


'       need to add Person third for referential integrity
        Person.FindFirst "SSN = "' & SSN & ""'"
          If Person.NoMatch Then
            With Person
             ' Add new record to end of Recordset object.
             .AddNew
             !fName = fName
             !lName = lName      ' Add data.
             !SSN = SSN
             !Paygrade = sRate
             !sex = sex
             !DOB = DateValue(bday)
             !DeptName = DEPARTMENT
             !commandName = commandName
             .Update              ' Save changes.
```

```vba
            End With
        End If


'       Now add results
        Dim COMMANDSSN As String
        COMMANDSSN = "SSN = '" & SSN & "' AND CommandName = '" &
commandName & "'"
        Results.FindFirst COMMANDSSN & " AND PRTNo = '" & myPRTNo &
"'"

        If Results.NoMatch Then
          With Results
            ' Add new record to end of Recordset object.
            .AddNew
            !SSN = SSN
            !commandName = commandName
            !PRTNo = myPRTNo
            !TestDate = prtdate
            !sitReach = sitReach
            !waist = waist
            !hip = Val(hip)
            !waiverCode = waiverCode
            !neck = neck
            !Height = sHeight
            !weight = weight
            !pushups = myPushups
            !curlups = mySitups
            !Type = RunSwim
            !Time = TimeValue(runTime)
            .Update              ' Save changes.
          End With
        End If



        fLen = fLen - 181


    Loop
```

84

```
CleanUp:
    UserMan.MousePointer = 0   'reset mouse
    Close #1
    MsgBox ("Merge was successful."), , "Merge"
End If
Command.Close
Dept.Close
Person.Close
PRT.Close
Results.Close
dbs.Close
Exit Sub
Didnotmerge:
Command.Close
Dept.Close
Person.Close
PRT.Close
Results.Close
dbs.Close
UserMan.MousePointer = 0   'reset mouse
MsgBox ("Merge was not successful."), , "Merge"
End Sub


Private Sub Command6_Click()
'On Error Resume Next
    Unload Command
    Unload DEPARTMENT
    Unload Filters
    Unload frmMenu


    Unload frmRESULTS


    Unload MainMenu
    Unload SERVICEMEMBER
    Unload UIC
    Unload UserMan
```

```
        Unload Welcome
        End
End Sub


Private Sub Command7_Click()


    SERVICEMEMBER.Show
End Sub


Private Sub Command8_Click()


    gsRecordsource = "RESULTS"
    Filters.Show
End Sub


Private Sub Command9_Click()
    UIC.Show
End Sub


Private Sub Form_Load()
    Data1.DatabaseName = App.Path & "\prt.mdb"
End Sub
```

'Results Screen

```
Dim commandName As String
Dim Personnel As Recordset
Dim Combo5Full As Boolean
Dim HasChanges As Boolean


Private Sub Combo5_Change()
  Combo5.Text = Personnel!!Name
End Sub


Private Sub Combo5_Click()
    Personnel.FindFirst "[Lname] = '" & Combo5.Text & "'"
    Dim SSN As String
```

86

```
        SSN = Personnel!SSN
        Data1.Recordset.FindFirst "[SSN] = '" & SSN & "'"
End Sub


Private Sub Command1_Click()

    Data1.Recordset.AddNew
    txtFields(2).Text = commandName

    MaskEdBox3.Mask = ""
    MaskEdBox3.Text = ""
    MaskEdBox3.Mask = "##/##/##"
    MaskEdBox4.Mask = ""
    MaskEdBox4.Text = ""
    MaskEdBox4.Mask = "##/##/##"
    MaskEdBox5.Mask = ""
    MaskEdBox5.Text = ""
    MaskEdBox5.Mask = "##/##/##"
    MaskEdBox6.Mask = ""
    MaskEdBox6.Text = ""
    MaskEdBox6.Mask = "##"
    MaskEdBox7.Mask = ""
    MaskEdBox7.Text = ""
    MaskEdBox7.Mask = "###"
    MaskEdBox13.Mask = ""
    MaskEdBox13.Text = ""
    MaskEdBox13.Mask = "##/##/##"
    MaskEdBox15.Mask = ""
    MaskEdBox15.Text = ""
    MaskEdBox15.Mask = "999"
    MaskEdBox16.Mask = ""
    MaskEdBox16.Text = ""
    MaskEdBox16.Mask = "999"
    MaskEdBox17.Mask = ""
    MaskEdBox17.Text = ""
    MaskEdBox17.Mask = "##:##:##"
```

```vb
        MaskEdBox11.Mask = ""
        MaskEdBox11.Text = ""
        MaskEdBox11.Mask = "##/##/##"
        MaskEdBox12.Mask = ""
        MaskEdBox12.Text = ""
        MaskEdBox12.Mask = "##/##/##"
        Data1.Recordset.Bookmark = Data1.Recordset.LastModified
End Sub


Private Sub Command2_Click()
'this may produce an error if you delete the last
    'record or the only record in the recordset
    With Data1.Recordset
        .Delete
        .MoveNext
        If .EOF Then .MoveLast
    End With
End Sub


Private Sub Command3_Click()
    Data1.UpdateRecord
    Data1.Recordset.Bookmark = Data1.Recordset.LastModified
End Sub


Private Sub Command4_Click()
    End
End Sub


Private Sub Command5_Click()
    Me.PrintForm
End Sub


Private Sub Data1_Error(DataErr As Integer, Response As Integer)
    This is where you would put error handling code
    'If you want to ignore errors, comment out the next line
    'If you want to trap them, add code here to handle them
```

```
            If Not DataErr = 383 Then
               MsgBox "Data error event hit err:" & Error$(DataErr)
            End If
            Response = 0   'throw away the error
         End Sub



         Private Sub Data1_Reposition()
            Screen.MousePointer = vbDefault
            On Error Resume Next

            'This will display the current record position
            'for dynasets and snapshots
            Data1.Caption = "Record: " & (Data1.Recordset.AbsolutePosition + 1)

            Personnel.FindFirst "[SSN] = '" & Data1.Recordset!SSN & "'"
            Combo5.Text = Personnel!lName

            If HasChanges Then
               Data6.Recordset.Requery
               Data6.Refresh
               HasChanges = False
            End If

            Dim t As Date
            t = Data1.Recordset!Time
            Dim m, s As Integer
            m = Minute(t)
            s = Second(t)
            Dim tStr As String
            tStr = IIf(m < 10, "0" & CStr(m), CStr(m)) & ":" & _
                IIf(s < 10, "0" & CStr(s), CStr(s))
            MaskEdBox17.Text = tStr

            Data2.Recordset.FindFirst "[SSN] = '" & Data1.Recordset!SSN & "'" & _
                       "AND [PRTNo] = '" & Data1.Recordset!PRTNo & "'"
```

89

```
If Data2.Recordset.NoMatch Then
    Label18.Caption = ""
    Label20.Caption = ""
End If
Data6.Recordset.FindFirst "[SSN] = '" & Data1.Recordset!SSN & "'" & _
            "AND [PRTNo] = '" & Data1.Recordset!PRTNo & "'"
If Data6.Recordset.NoMatch Then
    Label10.Caption = ""
    Label11.Caption = ""
    Label12.Caption = ""
    Label13.Caption = ""
    Label14.Caption = ""
    Label15.Caption = ""
    Label16.Caption = ""
    Label17.Caption = ""
End If


'If sex = male disable hip box

If SERVICEMEMBER!sex = "M" Then
    MaskEdBox10.Enabled = False
End If
    rs.Close
End Sub



Private Sub Data1_Validate(Action As Integer, Save As Integer)
    'This is where you put validation code
    'This event gets called when the following actions occur
    Select Case Action
        Case vbDataActionMoveFirst
        Case vbDataActionMovePrevious
        Case vbDataActionMoveNext
        Case vbDataActionMoveLast
        Case vbDataActionAddNew
        Case vbDataActionUpdate
```

90

```vb
        Case vbDataActionDelete
        Case vbDataActionFind
        Case vbDataActionBookmark
        Case vbDataActionClose
            Screen.MousePointer = vbDefault
    End Select
    Screen.MousePointer = vbHourglass
End Sub


Private Sub Form_Activate()


    If Not Combo5Full Then



        Data1.Recordset.MoveFirst
        Do While Not Data1.Recordset.EOF
            Personnel.FindFirst "[SSN] = '" & Data1.Recordset.SSN & "'"


            Combo5.AddItem Personnel!lName
            Data1.Recordset.MoveNext


        Loop


        Combo5Full = True
        Data1.Recordset.MoveFirst



    End If

End Sub


Private Sub Form_Load()
    Data6.DatabaseName = App.Path & "\prt.mdb"
    Data5.DatabaseName = App.Path & "\prt.mdb"
    Data2.DatabaseName = App.Path & "\prt.mdb"
```

```vb
Data1.DatabaseName = App.Path & "\prt.mdb"
Combo5Full = False
HasChanges = False
Dim wrkSpc As Workspace
Set wrkSpc = DBEngine.Workspaces(0)
Dim DB4 As Database
Set DB4 = wrkSpc.Databases(0)
Dim Command As Recordset
Set Command = DB4.OpenRecordset("COMMAND")
commandName = Command!commandName


Set Personnel = DB4.OpenRecordset("SERVICEMEMBER", dbOpenDynaset)



End Sub


Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub


Private Sub MaskEdBox15_Change()
    HasChanges = True
End Sub


Private Sub MaskEdBox15_LostFocus()
    Dim bm As Variant


    bm = Data1.Recordset.Bookmark


    Data1.Recordset.Edit
    Data1.Recordset!curlups = Val(MaskEdBox15.Text)
    Data1.Recordset.Update
    Data1.Recordset.Bookmark = bm


    Data6.Recordset.Requery
    Data6.Refresh
```

```vb
End Sub
Private Sub MaskEdBox16_Change()
  HasChanges = True
End Sub

Private Sub MaskEdBox16_LostFocus()
  Dim bm As Variant
  bm = Data1.Recordset.Bookmark
  Data1.Recordset.Edit
  Data1.Recordset!pushups = Val(MaskEdBox16.Text)
  Data1.Recordset.Update
  Data1.Recordset.Bookmark = bm
  Data6.Recordset.Requery
  Data6.Refresh
End Sub
Private Sub MaskEdBox17_Change()
  HasChanges = True
End Sub

Private Sub MaskEdBox17_LostFocus()
  Dim bm As Variant
  bm = Data1.Recordset.Bookmark
  Data1.Recordset.Edit
  Data1.Recordset!Time = CDate("00:" & MaskEdBox17)
  Data1.Recordset.Update
  Data1.Recordset.Bookmark = bm
  Data6.Recordset.Requery
  Data6.Refresh
End Sub
Private Sub MaskEdBox8_Change()
  HasChanges = True
End Sub

Private Sub MaskEdBox8_LostFocus()
  Dim bm As Variant
  bm = Data1.Recordset.Bookmark
```

```
        Data1.Recordset.Edit
        Data1.Recordset!Time = Val(MaskEdBox8.Text)
        Data1.Recordset.Update
        Data1.Recordset.Bookmark = bm
        Data2.Recordset.Requery
        Data2.Refresh
End Sub
Private Sub MaskEdBox9_Change()
    HasChanges = True
End Sub

Private Sub MaskEdBox9_LostFocus()
    Dim bm As Variant
    bm = Data1.Recordset.Bookmark
    Data1.Recordset.Edit
    Data1.Recordset!Time = Val(MaskEdBox9.Text)
    Data1.Recordset.Update
    Data1.Recordset.Bookmark = bm
    Data2.Recordset.Requery
    Data2.Refresh
End Sub
Private Sub MaskEdBox10_Change()
    HasChanges = True
End Sub

Private Sub MaskEdBox10_LostFocus()
    Dim bm As Variant
    bm = Data1.Recordset.Bookmark
    Data1.Recordset.Edit
    Data1.Recordset!Time = Val(MaskEdBox10.Text)
    Data1.Recordset.Update
    Data1.Recordset.Bookmark = bm
    Data2.Recordset.Requery
    Data2.Refresh
End Sub
```

'Forms Screen

```vb
Private Sub Command1_Click()
  End
End Sub




Private Sub optforms_Click(Index As Integer)

  Select Case Index
    Case 0
      'do nothing
    Case 1      'Page 13 One/Two Failure
      crpFormPage13One.Show
    Case 2      'Page 13 Three Failure
      crpFormPage13.Show
    Case 3      'Scoring Sheet
      crpScoringSheet.Show
    Case 4      'Advancement Report
     With Me.crptReports
        .ReportFileName = App.Path & "\prtrec.rpt"
        .WindowTitle = "Recommendations based on current prt"
        .WindowState = crptMaximized
        .Connect = "DSN=PRT"
        .Action = 1
      End With
    Case 5      'Recommendations based on 4 yr History
      With Me.crptReports
        .ReportFileName = App.Path & "\4yrrec.rpt"
        .WindowTitle = "Recommendations based on 4 year History"
        .WindowState = crptMaximized
        .Connect = "DSN=PRT"
        .Action = 1
```

```vb
        End With
     Case 6    'LON
        crpFormLON.Show


     Case 7 'Notice of CDCP
        With Me.crptReports
           .ReportFileName = App.Path & "\cdcp.rpt"
           .WindowTitle = "Recommendations based on current prt"
           .WindowState = crptMaximized
           .Connect = "DSN=PRT"
           .Action = 1
        End With

     Case 8   'Medical Review Board
        crpFormReferral.Show
     Case 9    'Award
        crpAward.Show
     Case Else
        ' no more at this time
     End Select

End Sub
Private Sub Form_Load()
   optforms.Item(0).Value = False
End Sub
-------------------------------------------------------------------------------------------

'Blank PRT Screen
Dim commandName As String
Dim Personnel As Recordset

Private Sub Command3_Click()
   End
End Sub
```

```vb
Private Sub Form_Load()
    Data1.DatabaseName = App.Path & "\prt.mdb"
    Data2.DatabaseName = App.Path & "\prt.mdb"
    Data3.DatabaseName = App.Path & "\prt.mdb"
    Dim wrkSpc As Workspace
    Set wrkSpc = DBEngine.Workspaces(0)
    Dim DB4 As Database
    Set DB4 = wrkSpc.Databases(0)

    Set Personnel = DB4.OpenRecordset("SERVICEMEMBER", dbOpenDynaset)


End Sub



Private Sub Command1_Click()
    Dim PrtNum As String
    Dim NewTestDate As Date

    On Error GoTo DateErrorMsg
    NewTestDate = CDate(meditTestDate.Text)
    PrtNum = Text1.Text

    On Error GoTo DBAddErrorMsg
    With Data2.Recordset

        .AddNew
        .PRTNo = PrtNum
        .commandName = commandName
        .TestDate = NewTestDate
        On Error GoTo DBUpdateErrorMsg
        .Update
    End With


    Rem Now add code to insert new blank records into
    Rem the results database for this PRT and all
    Rem individuals in the command that have not checked out
```

```vb
    Set Personnel = Me.Data1.Database.OpenRecordset("Servicemember")

    Do While Not Personnel.EOF
      If IsNull(Personnel!CHECKEDOUT) Then

        On Error GoTo DBAddErrorMsg
        With Data3.Recordset
          .AddNew
          .SSN = Personnel!SSN
          .PRTNo = PrtNum
          .TestDate = NewTestDate
          .commandName = commandName


          On Error GoTo DBUpdateErrorMsg
          .Update
        End With
      End If
      Personnel.MoveNext
    Loop
    Personnel.Close
    Set Personnel = Nothing


    frmBlankPrt.Hide


    Exit Sub


DateErrorMsg:
  MsgBox "Invalid Date format, please use mm/dd/yy", , "Date Error"
  Exit Sub
DBAddErrorMsg:
  MsgBox "Could not add new record"
  Exit Sub
DBUpdateErrorMsg:
  MsgBox "Could not update new record: " & Err.Description, , "Update Error"
  Exit Sub
```

```
        End Sub


        Private Sub Command2_Click()
            frmBlankPrt.Hide
        End Sub


        Private Sub Form_Activate()
         Data1.Recordset.MoveFirst
         commandName = Data1.Recordset!commandName
         Text1 = ""
         meditTestDate.Mask = ""
         meditTestDate.Text = ""
         meditTestDate.Mask = "##/##/##"


        End Sub


        Private Sub Form_Unload(Cancel As Integer)
            If Not (Personnel Is Nothing) Then
                Personnel.Close
                Set Personnel = Nothing
            End If
        End Sub


        --------------------------------------------------------------------------------------------------------

        'Filters Screen
        Dim sFilterStr As String
        Dim deptStr As String



        Private Sub Combo1_Click()
         Dim db As Database

          Select Case Combo1.ListIndex
            Case 0
                'All enlisted records
                Data1.Caption = "Enlisted"
```

```
                Me.MousePointer = vbHourglass
                On Error GoTo comboErr
                With Me.Data1
                   'Set db = .Database


                   .RecordSource = "SELECT * FROM  ENLISTEDRESULTS_TEST
WHERE PRTNO = '" & sFilterStr & "'"
                      .Refresh
                End With
                Me.MousePointer = vbDefault


             Case 1
                'All officers records
                Data1.Caption = "Officers"
                With Me.Data1
                   .RecordSource = "SELECT * FROM  OFFICERRESULTS_TEST
WHERE PRTNO = '" & sFilterStr & "'"
                      .Refresh
                End With
             Case 2
                'All records by selected department
                deptStr = UCase(InputBox("Enter Dept. Ex.ADMIN:"))
                If Len(sFilterStr) = 0 Then Exit Sub


                With Me.Data1
                   .RecordSource = "SELECT * FROM  FAILEDBYDEPT  WHERE
DEPTNAME = '" & deptStr & "'"
                      .Refresh
                End With
             Case 3
                'Who failed
                Data1.Caption = "Who Failed?"
                With Me.Data1
                .RecordSource = "SELECT * FROM WHOFAILED WHERE PRTNo = '"
& sFilterStr & "'"
                      .Refresh
```

```
            End With

        End Select


        Exit Sub
    comboErr:
        Screen.MousePointer = vbDefault
        MsgBox "Error:" & Err & " " & Err.Description, , "Combo Box Error"
        Exit Sub
    FilterErr:
        Screen.MousePointer = vbDefault
        MsgBox "Error:" & Err & " " & Err.Description, , "Filter Error"
        Exit Sub
    End Sub



    Private Sub Command1_Click()
        End
    End Sub

    Private Sub Form_Load()
        Data1.DatabaseName = App.Path & "\prt.mdb"
        On Error GoTo LoadErr

        sFilterStr = UCase(InputBox("Enter PRTNo. Ex.APR98:"))
        If Len(sFilterStr) = 0 Then Exit Sub

        With Me.Data1
            .RecordSource = "SELECT * FROM RESULTS WHERE PRTNO = '" &
    sFilterStr & "'"
            .Refresh
        End With

        Exit Sub
```

```
LoadErr:
    MsgBox "Err:" & Err & " " & Err.Description, , "Load Error"
    Unload Me



End Sub
```

```
'Department Screen
Dim commandName As String
Private Sub Form_Activate()
    Data2.Recordset.MoveFirst
    commandName = Data2.Recordset!commandName
End Sub
Private Sub Command1_Click()
    Data1.Recordset.AddNew
    Text2.Text = commandName
    Data1.Recordset.Bookmark = Data1.Recordset.LastModified

End Sub


Private Sub Command2_Click()
    'this may produce an error if you delete the last
    'record or the only record in the recordset
    With Data1.Recordset
        .Delete
        .MoveNext
        If .EOF Then .MoveLast
    End With
End Sub


Private Sub Command3_Click()
    Data1.UpdateRecord
    Data1.Recordset.Bookmark = Data1.Recordset.LastModified
End Sub
```

```vb
Private Sub Data1_Error(DataErr As Integer, Response As Integer)
    'This is where you would put error handling code
    'If you want to ignore errors, comment out the next line
    'If you want to trap them, add code here to handle them
    MsgBox "Data error event hit err:" & Error$(DataErr), , "Data Error"
    Response = 0  'throw away the error
End Sub


Private Sub Data1_Reposition()
    Screen.MousePointer = vbDefault
    On Error Resume Next
    'This will display the current record position
    'for dynasets and sanpshots
    Data1.Caption = "Record: " & (Data1.Recordset.AbsolutePosition + 1)
    'for the table object you must set the index property when
    'the recordset gets created and use the following line
    'Data1.Caption    =    "Record:    "    &    (Data1.Recordset.RecordCount    *
(Data1.Recordset.PercentPosition * 0.01)) + 1
    End Sub


Private Sub Data1_Validate(Action As Integer, Save As Integer)
    'This is where you put validation code
    'This event gets called when the following actions occur
    Select Case Action
        Case vbDataActionMoveFirst
        Case vbDataActionMovePrevious
        Case vbDataActionMoveNext
        Case vbDataActionMoveLast
        Case vbDataActionAddNew
        Case vbDataActionUpdate
        Case vbDataActionDelete
        Case vbDataActionFind
        Case vbDataActionBookmark
        Case vbDataActionClose
            Screen.MousePointer = vbDefault
    End Select
```

```
    Screen.MousePointer = vbHourglass
End Sub


Private Sub Form_Load()
    Data1.DatabaseName = App.Path & "\prt.mdb"
    Data2.DatabaseName = App.Path & "\prt.mdb"
End Sub


Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
```

'Crystal Reports Screens
```
Dim Report As New CrystalReport5


Private Sub Form_Load()
CRViewer1.ReportSource = Report
CRViewer1.ViewReport
End Sub


Private Sub Form_Resize()
CRViewer1.Top = 0
CRViewer1.Left = 0
CRViewer1.Height = ScaleHeight
CRViewer1.Width = ScaleWidth
End Sub
```

LIST OF REFERENCES

1. CINCPACFLT 300944Z Mar 97.

2. Clemins, A. IT21: Moving to the 3rd Stage. In *Proceedings*, pp. 51-54, May 1997.

3. OPNAVINST 6110.1E of March 23 1988.

4. Brown, W.L. 1988 PRT Software Package. User Manual, May 1988.

5. Burlage, B. New Standards and Second Chances: The PRT you haven't seen. In *Navy Times*, March 16, 1998.

6. Clemins, A. IT21: The Path to Information Superiority. [Http://www.chips.navy.mil/chips/archives/97jul/file1.htm].

7. Gissel, M.A. Command Presentation for NAS South Weymouth on the Command Fitness Program.

8. Gissel, M.A. Command Fitness Program Relational Management Database. Users Guide, August 1994.

9. Elmasri, Ramez. *Fundamentals of database design*, Addison-Wesley Publishing Company, 1994.

10. McManus, Jeffrey P. *Database Access with Visual Basic*, Sams Publishing, 1998.

11. The U.S. Navy Circumference Method, [Http://tempura.cs.uq.edu.au/personal/Weights/bodyfat.html].

12. Microsoft Visual Basic 5.0 Programmers Guide, Microsoft Press, 1997.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center..2
    8725 John J. Kingman Road, Ste 0944
    Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
    Naval Postgraduate School
    411 Dyer Rd.
    Monterey, CA 93943-5101

3. Dr.D.C. Boger...1
    Department of Computer Science
    Naval Postgraduate School
    Monterey, CA 93943-5100

4. Dr. C. Thomas Wu..1
    Department of Computer Science
    Naval Postgraduate School
    Monterey, CA 93943-5100

5. LCDR Christopher S. Eagle...1
    Department of Computer Science
    Code CS/CE
    Naval Postgraduate School
    Monterey, CA 93943-5100

6. Department of Navy 1
    Bureau of Naval Personnel
    PERS-601
    2 Navy Annex
    Washington, DC 20370-6010

7. Frances M. Quinones..1
    159 Padre Tullio Drive
    San Ysidro, CA 92173